Analyse der Generalisierbarkeit von maschinell gelernten Algorithmen in Fahrerassistenzsystemen

Vom Fachbereich Maschinenbau an der
Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
eingereichte

Dissertation

vorgelegt von

Maren Henzel, M. Sc. (geb. Graupner)
aus Büdingen

Berichterstatter:	Prof. Dr. rer. nat. Hermann Winner
Mitberichterstatter:	Prof. Dr. techn. Johannes Fürnkranz

Tag der Einreichung:	20.05.2019
Tag der mündlichen Prüfung:	17.09.2019

Darmstadt 2019

D 17

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Fachgebiet Fahrzeugtechnik (FZD) der Technischen Universität Darmstadt. Die Inhalte dieser Dissertation resultieren aus dem Forschungsprojekt PRORETA 4, das in Kooperation mit der Continental AG durchgeführt wurde.

Ein besonderes Dankeschön spreche ich meinem Doktorvater Herrn Prof. Dr. rer. nat. Hermann Winner aus. Die vielen inhaltlichen Diskussionen, sein Vertrauen in mich und meine Arbeit sowie die gewährten Freiheiten haben einerseits einen maßgeblichen Beitrag zum Gelingen dieser Arbeit beigetragen und andererseits mich und meine persönliche Entwicklung geprägt.

Prof. Dr. techn. Johannes Fürnkranz, Leiter der Knowledge Engineering Group der TU Darmstadt, danke ich herzlich für die Übernahme des Korreferats sowie für den fachlichen Austausch in den PRORETA 4 Projektbesprechungen und darüber hinaus.

Bei dem Projektpartner Continental AG möchte ich mich für die Finanzierung, die Unterstützung im Projekt aber auch die stets angenehme Zusammenarbeit bedanken. Stellvertretend für zahlreiche weitere Mitarbeiter der beteiligten Divisionen, Chassis & Safety und Interior, danke ich an dieser Stelle auch ganz persönlich den Haupt-Betreuern des Projekts Dr. Benedikt Lattke und Maximilian Höpfl.

Nicht nur die Betreuung des Projekts trug maßgeblich zum Erreichen des vorliegenden Meilensteins bei, sondern auch die Zusammenarbeit mit meinen Teamkollegen Hien Dang, Stefan Luthardt und Julian Schwehr. Für die herausragende Zusammenarbeit, den ständigen fachlichen Austausch sowie die entstandene Freundschaft möchte ich mich herzlich bedanken.

Des Weiteren bedanke ich mich bei allen Mitarbeitern des Fachgebiets Fahrzeugtechnik, insbesondere auch der Werkstatt und dem Sekretariat. Die Arbeitsatmosphäre, der Zusammenhalt und der fachliche sowie private Austausch werden mir in guter Erinnerung bleiben. Die entstandenen Freundschaften werden mich glücklicherweise auch nach dieser Zeit auf meinem weiteren Weg begleiten.

Nicht zuletzt danke ich meiner Familie, die mich während meiner gesamten Ausbildung unterstützt haben und mir in jeder Lebenslage mit Rat und Rückhalt zur Seite stehen. Sie ermöglichen es mir, mein Leben nach meinen Interessen, Wünschen und Entscheidungen zu führen. Mein ganz besonderer Dank gilt meinem Mann Patrick für sein Verständnis und die liebevolle Unterstützung und meinem Sohn Maro, der mir immer zeigt, was im Leben wirklich zählt.

Maren Henzel Büdingen, Mai 2019

Inhaltsverzeichnis

Abkürzungen

Abkürzung	Beschreibung
ACC	Adaptive Cruise Control
ANN	Artificial Neuronal Network
ASIL	Automotive Integrity Level
CDF	Kumulative Verteilungsfunktion
CNN	Convolutional Neuronal Network
FAS	Fahrerassistenzsystem
FMEA	Fehlermöglichkeits- und –einflussanalyse
FTA	Fault Tree Analysis
GMM	Gaussian Mixture Model
GSN	Goal Structuring Notation
HMM	Hidden Markov Model
MA	Gleitender Mittelwert
ML	Maschinelles Lernen
NN	Neuronales Netz / Neuronal Network
StVZO	Straßenverkehrs-Zulassungs-Ordnung
SVM	Support Vector Machine
TLE	Top Level Event
TMA	Triangularer gleitender Mittelwert
ÜaC	Übereinstimmung aller Clustervorhersagen
ÜvC	Übereinstimmung der Vorhersagen der verbleibenden Cluster

Formelzeichen und Indizes

Symbol	Einheit	Beschreibung
Acc	-	Genauigkeit / Accuracy
C	-	Cluster
FN	-	False Negative
FP	-	False Positive
k	-	Anzahl an Datenuntermengen / Partitionen
n	-	Anzahl
p	-	Wahrscheinlichkeit
TN	-	True Negative
TP	-	True Positive
x	differierend	Eingangsgröße
$\bar{x}$	-	Mittelwert der Eingangsgröße

Index	Beschreibung
init	Initialisierung
S	Erfolgreicher Testfall (success)
T	Testfall

Abbildungen und Tabellen

Kurzzusammenfassung

Maschinell gelernte Systeme finden immer häufiger Einsatz in Fahrerassistenzsystemen. Dabei zeichnen sich die eingesetzten Lernalgorithmen je nach Aufgabenkomplexität durch eine geringe Nachvollziehbarkeit der Vorhersagekriterien gewünschter Ausgangsgrößen aus. Darüber hinaus basiert Maschinelles Lernen auf einem induktiven Erkenntnisprozess in welchem nicht sichergestellt ist, dass die zur Vorhersage der Ausgangsgrößen erlernten Zusammenhänge allgemeingültig sind bzw. auf Kausalität anstelle von Korrelation basieren. Daher ist es möglich, dass die Generalisierbarkeit des gelernten Modells nicht für die Abdeckung des gesamten Betriebsbereichs ausreicht. Unter anderem hierdurch bedingt sind bestehende Möglichkeiten zur Erbringung eines Sicherheitsnachweises nicht anwendbar. Die bestehenden alternativen Ansätze zum Nachweis der Sicherheit schränken entweder die Leistungsfähigkeit der genutzten Modelle stark ein oder unterliegen der Annahme, dass alle inhärenten Fehler gelernter Modelle zuverlässig identifiziert werden. Ein Fehler liegt dabei in der bereits beschriebenen fehlenden Generalisierbarkeit. Um die Sicherheitsnachweise, die die Leistungsfähigkeit nicht einschränken, anzuwenden, ist daher zuvor u.a. die Analyse der Generalisierbarkeit des gelernten Modells durchzuführen.

Da hierzu keine systematische Methode identifiziert wurde, bildet deren Entwicklung nach der Analyse der bestehenden Sicherheitsnachweise den weiteren Fokus der vorliegenden Arbeit. Als Grundlage zur Entwicklung des Ansatzes werden die Ursachen der fehlenden Generalisierbarkeit mittels einer Fehlerbaumanalyse untersucht. Diese Ursachen gliedern sich in drei Kategorien hinsichtlich potentieller Vermeidungs- und Identifikationsmöglichkeiten. Aufbauend auf dieser Gliederung wird ein systematischer vierstufiger Ansatz zur Vermeidung bzw. Identifikation fehlender Generalisierbarkeit abgeleitet. Die ersten beiden Schritte des Ansatzes bestehen in der Vermeidung und der direkten Identifikation von Ursachen fehlender Generalisierbarkeit. Dadurch, dass jedoch nicht alle Ursachen vermeidbar oder deren Auftreten nicht direkt identifizierbar sind, werden im dritten und vierten Schritt das Vorliegen der Auswirkungen fehlender Generalisierbarkeit untersucht. Hierzu wird im dritten Schritt die Einhaltung funktionaler Anforderungen durch das gelernte Modell überprüft. Durch die nicht auszuschließende Unvollständigkeit dieser funktionalen Anforderungen besteht der vierte Schritt des Ansatzes aus der Überprüfung der Robustheit bzw. Sensitivität des Modells auf Veränderungen, die bei ausreichender Generalisierbarkeit des Modells zu keiner Änderung des funktionalen Verhaltens führen.

Der Ansatz wird prototypisch auf ein gelerntes Modell eines Fahrerassistenzsystems angewendet, um die praktische Anwendbarkeit des Ansatzes sowie den resultierenden Erkenntnisgewinn über die Generalisierbarkeit des Modells zu untersuchen. Die praktische Anwendbarkeit wird unter der Voraussetzung gezeigt, dass der Aufwand zum Training neuer Modelle und ggf. der Erhebung neuer Testdaten getragen wird. Darüber hinaus wird belegt, dass mit diesem Ansatz mögliche fehlende Generalisierbarkeit identifiziert werden

kann. Aus den Grenzen des Ansatzes ergeben sich die weiteren vorgestellten Forschungsfragen. Der hier vorgestellte Ansatz ergänzt die bestehenden Sicherheitsnachweise nun um einen für das Maschinelle Lernen spezifischen Baustein.

1 Einleitung

1.1 Motivation

Künstliche Intelligenz findet immer stärker Einzug in Fahrzeugsysteme. Vor allem im Infotainment-Bereich, aber auch im Rahmen von Fahrerassistenzsystemen (FAS) und bei der Umsetzung von (teil-)automatisiertem Fahren wird erwartet, dass maschinell gelernte Modelle vielfach die konventionell programmierten Algorithmen ersetzen. Dabei soll die Anzahl von Systemen, die sich Künstlicher Intelligenz bedienen, laut einer Schätzung bis zum Jahr 2025 auf 225 Millionen Fahrzeugsystem-Einheiten anwachsen.[1] Durch deren Einsatz ergeben sich Änderungen im Systementwicklungsprozess, da sich Lernalgorithmen dem Prinzip der Induktion bedienen.[2] Aus einer Datenmenge wird automatisiert ein Modell erstellt, wobei erwartet wird, dass das generierte Modell auch über die zur Verfügung gestellte Datenmenge hinaus eine generelle Gültigkeit besitzt. Im Gegensatz hierzu wurden FAS bisher anforderungsbasiert entwickelt und deren Funktionalität und Sicherheit anhand der Anforderungserfüllung bewertet. Die zur Erstellung der gelernten Modelle benötigten Daten werden beispielsweise aus Kundenfahrzeugen erhoben, wie im Fall von Tesla.[3] Neben Fragen zum Datenschutz, die sich hierdurch ergeben, stellt sich die Frage, ob diesen hiermit generierten Modellen im Sinne ihrer Sicherheit zu vertrauen ist.

Aufgrund ihrer Sicherheitsrelevanz sind FAS entsprechend der Vorgaben der ISO 26262[4] sowie der ISO/ PAS 21448[5] zu entwickeln. Jedoch widerspricht die Verwendung von Lernalgorithmen diesen Vorgaben in mehreren Punkten. Beispielsweise fordert die ISO 26262 eine anforderungsbasierte Entwicklung von Systemen und deren Submodulen, was, wie bereits erläutert, durch die Nutzung von Maschinellem Lernen (ML) nicht gegeben ist. Des Weiteren werden mögliche Fehler, die den Algorithmen u.a. durch das Nutzen des Induktions-Prinzips inhärent sind, nicht von diesen Vorgaben adressiert.[6] Die ML-inhärenten Fehler treten auch dann auf, wenn den resultierenden Modellen eine hohe Leistungsfähigkeit zugeschrieben wird. Dies liegt darin begründet, dass es lediglich möglich ist, die Leistungsfähigkeit der Modelle an den während der Entwicklung zur Verfügung stehenden

[1] Vgl. DeAmbroggi, L.: Artificial Intelligence Systems (2016).

[2] Vgl. Bergadano, F.: The Problem of Induction and Machine Learning (1991), S. 1074.

[3] Vgl. Marr, B.: The Amazing Ways Tesla Is Using Artificial Intelligence And Big Data (2018).

[4] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[5] ISO: ISO/ PAS 21448 (2019).

[6] Vgl. Salay, R. et al.: An Analysis of ISO 26262 (2017).

Daten zu messen. Ändern sich beispielsweise die Bedingungen, unter denen das Modell eingesetzt wird, wird hierdurch ebenfalls der Raum an möglichen Eingangsdaten beeinflusst. Die in der Entwicklung ermittelte Leistungsfähigkeit ist für diesen neuen Datenraum, je nach Höhe der Generalisierbarkeit der gelernten Zusammenhänge, nicht mehr uneingeschränkt gegeben. Ohne den Nachweis der Generalisierbarkeit von gelernten Modellen ist daher der Nachweis der Sicherheit von Fahrerassistenzsystemen nicht möglich, da trotz hoher Leistungsfähigkeit des Algorithmus Fehler verbleiben können.

Basierend auf vielfältigen Aussagen bezüglich bestehender Problemstellungen im Nachweis der Sicherheit von Maschinellem Lernen in FAS von, unter anderem, Salay und Czarnecki[7], Burton[8], Faria[9] und Rudolph[10] lautet die initale Forschungsfrage der vorliegenden Arbeit „*Welche Herausforderungen bestehen im Nachweis der Sicherheit von ML in FAS?*“. Bestehende Konzepte zum Beweis der Sicherheit von ML werden hinsichtlich ihrer Defizite untersucht und anschließend ggf. weitere Analysen der identifizierten Defizite durchgeführt.

1.2 Forschungsprozess und Struktur der Arbeit

Um die Forschungsfrage, welche Herausforderungen im Nachweis der Sicherheit von ML in FAS bestehen, zu diskutieren, werden in Kapitel 2 zunächst die Grundlagen zur Absicherung von Fahrerassistenzsystemen und Maschinellem Lernen gelegt.

Da sich die verschiedenen Arten des ML in ihrem Entwicklungsprozess unterscheiden und hiervon ein Sicherheitsnachweis abhängig ist, wird in Kapitel 3 zunächst der Stand der Wissenschaft und Technik der aktuell eingesetzten Arten gelernter Modelle in Fahrerassistenzsystemen analysiert. Für die im Fokus befindlichen Algorithmenarten werden bestehende Ansätze und Methoden zur Erbringung eines Sicherheitsnachweises vorgestellt. Diese werden analysiert, um festzustellen, ob und welche Defizite bestehen oder hierdurch bereits die Frage nach den bestehenden Herausforderungen mit „keine“ beantwortet wird.

Aufgrund der aus Kapitel 3 resultierenden, bestehenden Forschungsfrage nach einem systematischen Ansatz zur Analyse möglicher fehlender Generalisierbarkeit in gelernten Modellen wird ein ebensolcher in Kapitel 5 entwickelt. Hierzu werden in Kapitel 4 zunächst die Ursachen fehlender Generalisierbarkeit analysiert, um basierend hierauf mögliche Begegnungsmaßnahmen abzuleiten. Die Frage nach der praktischen Anwendbarkeit des Ansatzes wird in Kapitel 6 beantwortet, um die hieraus gewonnenen Erkenntnisse über die

[7] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018).

[8] Burton, S.; Bürkle, L.: Making the Case for Safety of Machine Learning (2017).

[9] Faria, J.: Machine Learning Safety (2018).

[10] Rudolph, A. et al.: A consistent safety case argumentation for artificial intelligence (2018).

Grenzen der Methodik in Kapitel 7 zu diskutieren. In diesem Kapitel werden darüber hinaus die weithin bestehenden Herausforderungen abgeleitet. Mit einer Zusammenfassung und einem Ausblick schließt die vorliegende Arbeit ab (Kapitel 8). Die vorgestellte Struktur ist in Abbildung 1-1 verdeutlicht.

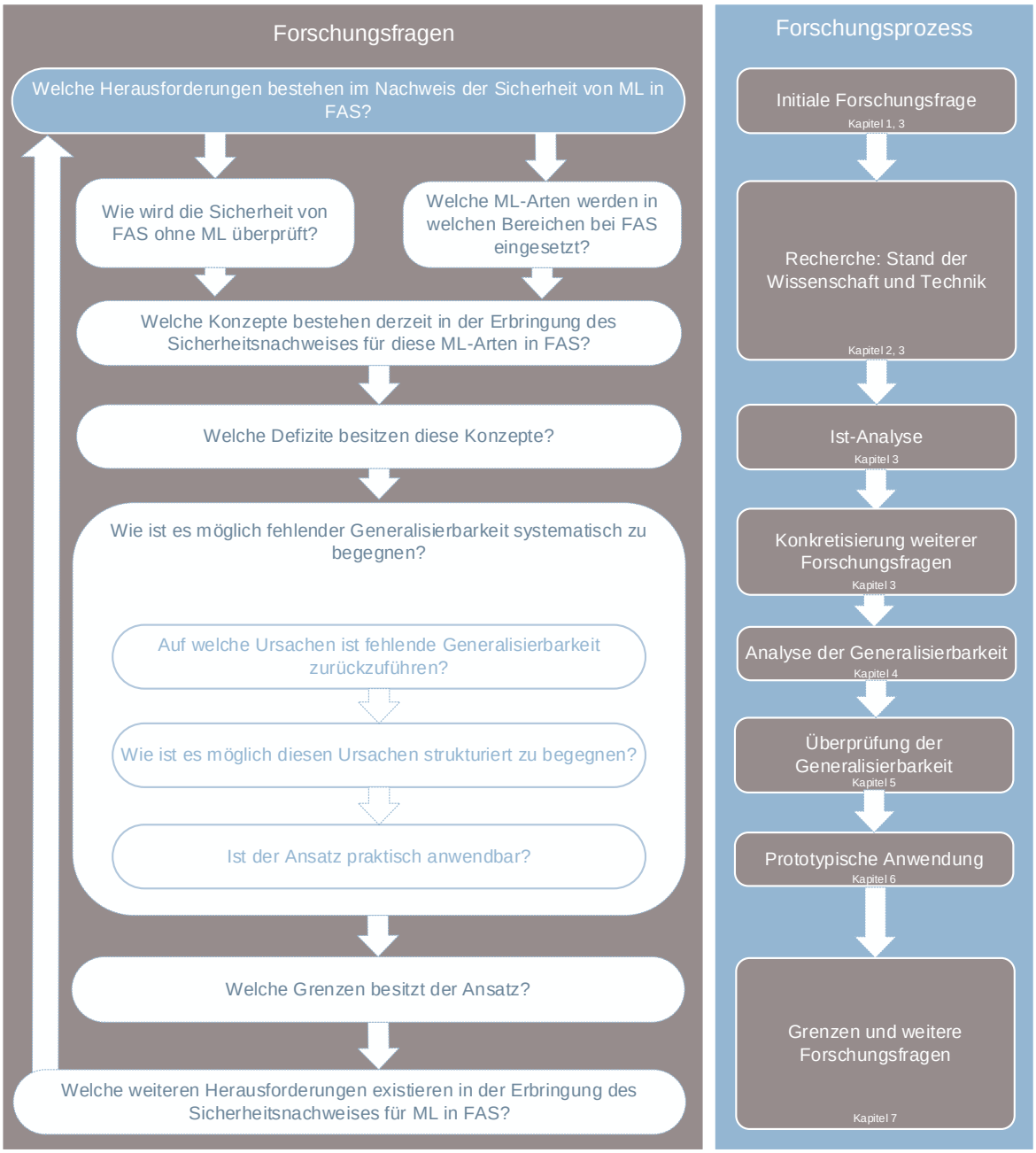


Abbildung 1-1: Forschungsfragen (links) und Forschungsprozess (rechts)

2 Grundlagen

Um die Frage, ob und unter welchen Voraussetzungen gelernten Modellen in Fahrerassistenzsystemen zu vertrauen ist, zu untersuchen, wird zunächst in Unterkapitel 2.1 der Stand der Praxis in der Erbringung des Sicherheitsnachweises von FAS ohne Einsatz von ML dargestellt und hierdurch der Frage „*Wie wird die Sicherheit von FAS ohne ML überprüft?*“ (siehe Unterkapitel 1.2) beantwortet. Nach der Einführung in die Grundlagen über den Entwicklungsprozess sowie die verschiedenen Arten des maschinellen Lernens in Unterkapitel 2.2 wird in Unterkapitel 2.3 auf die Frage „*Welche ML-Arten werden in welchen Bereichen bei FAS eingesetzt?*“ mit der Analyse des Standes der Wissenschaft und Technik in diesem Themenfeld eingegangen.

2.1 Sicherheitsnachweis von Fahrerassistenzsystemen

Ausgehend von geltenden Gesetzen in Deutschland werden zunächst die rechtlichen Grundlagen zur Erbringung des Sicherheitsnachweises dargestellt (Abschnitt 2.1.1), um anschließend in jeweils einem eigenen Abschnitt auf zwei technischen Standards einzugehen, die es zu erfüllen gilt. Welche Methoden derzeit genutzt werden, um die Erfüllung dieser Standards zu beweisen, wird in Abschnitt 2.1.4 vorgestellt. In diesem Abschnitt wird ebenfalls eine zusammenfassende Antwort auf die Forschungsfrage „*Wie wird die Sicherheit von FAS ohne ML überprüft?*“ gegeben.

Die nachfolgenden Textpassagen sind teilweise wörtlich der Veröffentlichung Henzel et al.[11] entnommen.

2.1.1 Rechtliche Grundlagen

Die grundsätzliche Bedingung für die Teilnahme am Straßenverkehr mit einem Fahrzeug stellt das Vorliegen einer gültigen Zulassung basierend auf einer Typengenehmigung für Fahrzeuge dar.[12] Aus der Typengenehmigung für Fahrzeuge leiten sich Anforderungen an das Fahrzeug und dessen Systeme ab, die in den jeweiligen nationalen Verordnungen oder Gesetzen festgelegt sind. In Deutschland ist dies die Straßenverkehrs-Zulassung-Ordnung (StVZO). Es existiert kein direkter Bezug zu technischen Vorschriften, Standards oder

[11] Henzel, M. et al.: Herausforderungen in der Absicherung von FAS (2017).

[12] Vgl. Bundesministerium der Justiz und für Verbraucherschutz: FZV (2011).

Normen innerhalb dieser Verordnung. Die Verbindung zwischen Normen und der StVZO wird durch das Produktsicherheitsgesetz[13] und das Produkthaftungsgesetz[14] geschaffen, da hier der Hersteller angehalten wird, nachzuweisen, dass sein Produkt mindestens den erforderlichen Sicherheitsstandard nach dem Stand von Wissenschaft und Technik erfüllt. Dieser Standard ist durch Normen und sonstige Richtlinien definiert.[15]

Eine Übersicht der Normen und Richtlinien zum Nachweis der Sicherheit in FAS ist in Weitzel et al.[15] aus dem Jahr 2014 zu finden. Hierbei werden der „Code of Practice“[16] sowie die ISO 26262[17], die sich mit dem Vorgehen zur Erreichung funktionaler Sicherheit beschäftigt, als Normen und Vorschriften mit hoher Relevanz für die Absicherung identifiziert. Funktionale Sicherheit bezieht sich auf die Sicherheit von elektronischen/ elektrischen Komponenten im Fahrzeug. Im „Code of Practice“ ist der Stand der Technik zur Bewertung der Kontrollierbarkeit von Fahrerassistenzsystemen mit Umfeldwahrnehmung im Fehlerfall zusammengefasst. Dieser bezieht sich daher auf die Umsetzung der Norm ISO 26262.[15]

Als Ergänzung zur ISO 26262 für einen ganzheitlichen Blick auf die Sicherheit von FAS wurde ab 2014 eine weiterer Standard entwickelt: die ISO/ PAS 21448[18], welche sich mit der Gewährleistung einer sicheren Sollfunktion beschäftigt. Die Bezeichnung „PAS“ steht für „Public Avaliable Specification“, da das Dokument zwar eine öffentliche Anforderung darstellt, jedoch bisher keine internationale Norm. Die in der PAS festgehaltenen Anforderungen sind der Konsens einer ISO-externen Organisation oder Arbeitsgruppe. Durch eine PAS wird auf dringende Marktbedürfnisse reagiert.[19]

Zusätzlich existiert die ISO/ SAE CD 21434[20], die zur Sicherstellung der Angriffssicherheit, also dem Schutz von Gefahren gegenüber der Systemumwelt, dient.[21] „CD“ bezeichnet einen „Committee Draft“, der den Mitgliedern des Ausschusses zur Kommentierung zur Verfügung gestellt wird. Hieraus soll ein technischer Konsens entwickelt werden.[22] Die Zusammenhänge zwischen den Themengebieten der Standards bzw. Dokumenten ist mit Abbildung 2-1 dargestellt. Die Achsen des Diagramms stellen dabei unterschiedliche Dimensionen der Sicherheit dar, die jeweils von einem der Dokumente adressiert werden.

---

[13] Bundesministerium der Justiz und für Verbraucherschutz: ProdSG (2011).

[14] Bundesministerium der Justiz und für Verbraucherschutz: ProdHaftG (1989).

[15] Vgl. Weitzel, A. et al.: Absicherungsstrategien für Fahrerassistenzsysteme (2014).

[16] Vgl. Brockmann, M.: Code of Practice for the Design and Evaluation of ADAS (2009).

[17] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[18] ISO: ISO/ PAS 21448 (2019).

[19] Vgl. ISO: Deliverables (2019).

[20] ISO: ISO/ SAE CD 21434 (2019).

[21] Vgl. Schnieder, L.; Hosse, R. S.: Leitfaden Safety of the Intended Functionality (2019), S. 3f.

[22] Vgl. ISO: My ISO job (2018), S. 25.

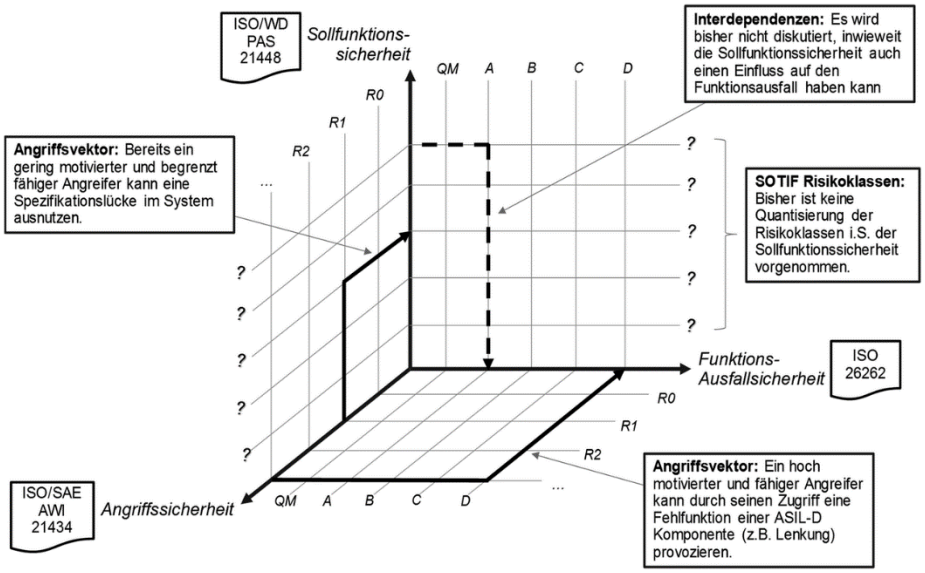


Abbildung 2-1: Achsen der Sicherheit für sicherheitsrelevante FAS[23]

Die ISO/ SAE CD 21434 wird im Folgenden nicht näher vorgestellt, da die Thematik der Angriffssicherheit zwar im FAS-Kontext relevant ist, die Gefahren von außen in der vorliegenden Arbeit jedoch ausgeschlossen werden. Es werden zunächst Funktionalitäten fokussiert werden, die keinen Angriff von außen ermöglichen. Daher wird beispielsweise eine Update-Fähigkeit der Fahrzeugfunktionen, wie sie einige Fahrzeughersteller (bspw. Tesla[24]) bereits anbieten, nicht betrachtet.

2.1.2 ISO 26262

Die ISO 26262[25] fordert die Einhaltung von Vorgaben an die Entwicklung sicherheitskritischer Elektronik- und Elektrik-Komponenten und Systeme im Fahrzeug bis 3,5 t über den gesamten Produktlebenszyklus. Ziel der Norm ist es, funktionale Sicherheit herzustellen, d.h. systematische und zufällige Fehler der Funktion zu beherrschen. Das in der Norm beschriebene Vorgehen orientiert sich am V-Modell (siehe Abbildung 2-2), welches aus der Produktenwicklung bekannt ist.[26] Zu Beginn des Systemlebenszyklus sind Anforderungen zu definieren, deren Umsetzung bis auf Modulebene weiterverfolgt wird. Die Erfüllung der Anforderungen wird jeweils innerhalb der Entwicklungsstufen mit Testfällen überprüft.

[23] Schnieder, L.; Hosse, R. S.: Leitfaden Safety of the Intended Functionality (2019), S. 4.

[24] Tesla: Software-Updates (2019).

[25] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[26] Vgl. Wilhelm, U. et al.: Functional Safety of Driver Assistance Systems and ISO 26262 (2016), S. 111.

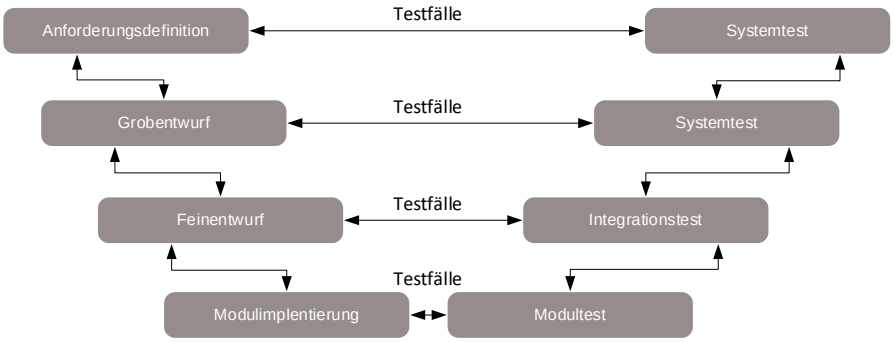


Abbildung 2-2: V-Modell in Anlehnung an Balzert[27]

Diese Anforderungen adressieren dabei auch die zu erreichende Sicherheit des Systems, welche durch Sicherheitsziele abgebildet wird. Zur Bestimmung dieser Sicherheitsziele wird zunächst eine Gefahren- und Risikoanalyse basierend auf einer vorläufigen Systembeschreibung durchgeführt. Aus dieser Risikoanalyse werden Sicherheitsanforderungen abgeleitet, denen Grenzwerte für akzeptierte Ausfallwahrscheinlichkeiten (sog. Automotive Safety Integrity Level, ASIL) zugeordnet sind. Aus diesen Anforderungen leiten sich anschließend die systemspezifischen Sicherheitsziele ab.[28] Basierend auf dem zugeordneten ASIL der betreffenden Systemkomponente werden neben den einzuhaltenden Sicherheitszielen Prozesse, die zur Produktentwicklung einzuhalten sind, sowie Methoden zur Überprüfung der Systemkomponenten, wie bestimmte Softwaretools, vorgeschrieben.

2.1.3 ISO/ PAS 21448

Die ISO/ PAS 21448[29] besitzt zum Ziel, die Gebrauchssicherheit bzw. die Sicherheit der Sollfunktion herzustellen. Die Thematik ist unter dem Begriff „Safety of the Intended Function“ bzw. SOTIF bekannt. Auch wenn ein System nach ISO 26262 als frei von Fehlern im Sinne des Funktionsausfalls gilt, ist es dennoch möglich, dass Sicherheitsverletzungen durch bisher nicht bekanntes Systemverhalten hervorgerufen werden. Von einer fehlerhaften Sollfunktion wird gesprochen, wenn das Verhalten des Systems nicht ausreichend bekannt oder spezifiziert ist. Hierdurch bedingt gilt es das mögliche Systemverhalten, vor allem das unsichere Systemverhalten, auch bei bislang unbekannten Anwendungsszenarien besser einzuschätzen. Inakzeptierte Personengefährdung ist auszuschließen, auch bei einem (vorhersehbaren) Fehlgebrauch des Systems. Die Ergebnisse des Standards, wie Anforderungen und zu überprüfende Anwendungsszenarien, werden direkt an die Konzeptphase der ISO 26262 übergeben, da die keinen eigenen Engineering Prozess vorsieht.[30]

---

[27] Balzert, H.: Lehrbuch der Softwaretechnik (1998).

[28] Eine beispielhafte Umsetzung der Risikoanalyse und ASIL-Zuordnung ist in Anhang A.2.1 und A.2.2 gegeben.

[29] ISO: ISO/ PAS 21448 (2019).

[30] Vgl. Schnieder, L.; Hosse, R. S.: Leitfaden Safety of the Intended Functionality (2019), S. 7ff.

2.1.4 Bestehende Methoden zum Nachweis der Sicherheit

Zu Beginn heutiger FAS-Sicherheitsnachweisstrategien wird analog zu den Vorgaben der ISO 26262 eine Risiko- bzw. Gefahrenanalyse durchgeführt, aus der sich Sicherheitsziele ableiten. Diese resultieren in einem funktionalen und technischen Sicherheitskonzept. Es besteht die Notwendigkeit, die Erfüllung dieser Ziele bzw. der hieraus resultierenden Sicherheitsanforderungen der Sicherheitskonzepte nachzuweisen. Zusätzlich zu denen durch die ISO 26262 hervorgehenden Sicherheitsanforderungen, ist im Rahmen der ISO/ PAS 21448 ebenfalls eine Risiko- bzw. Gefahrenanalyse mit Fokus auf die Gebrauchssicherheit durchzuführen, woraus weitere Anforderungen resultieren. Diese werden innerhalb der Phase der Erstellung des Sicherheitskonzepts der ISO 26262 übergeben. Mit steigendem Umsetzungsgrad des Systems ist es möglich, konkretere Risikobewertungsverfahren anzuwenden. Eine Auflistung sowie ein Vergleich verschiedener Verfahren ist bei Berg et al.[31] zu finden.

Um die Erfüllung der Sicherheitsziele und des zuvor definierten Funktionsumfangs nachzuweisen, existieren verschiedene Verfahren. Es besteht einerseits die Möglichkeit, durch analytische Beweisführung die Erfüllung der Anforderungen zu zeigen. Dies erfordert das Verstehen des betrachteten Systems und der zugehörigen Systemgrenzen. Andererseits ist es möglich die Anforderungserfüllung durch zuvor definierte Testfälle zu beweisen. Die verschiedenen Varianten, die zur Abprüfung der Testfälle herangezogen werden, unterscheiden sich in ihrem Abbildungsgrad der Realität bzw. realer Komponenten und der realen Umwelt. Die zur Testumsetzung eingesetzten Verfahren sind Berg et al.[31] und Rüger et al.[32] zu entnehmen. Die Herausforderung bei allen Testverfahren liegt in der Definition der korrekten Testfälle, um einerseits alle Aspekte der Anforderungen abzuprüfen und anderseits die Anzahl der benötigten Testfälle möglichst gering zu halten. Aufgrund des zu erwartenden zeitlichen und finanziellen Aufwands der Testdurchführung, wird empfohlen, die analytische Beweisführung dem Abtesten der Anforderungen vorzuziehen.

Zusammengefasst ist daher die Frage „*Wie wird die Sicherheit von FAS ohne ML überprüft?*“ wie folgt zu beantworten:

> Nach Identifikation möglicher Gefahren werden hierauf basierend Sicherheitsanforderungen aufgestellt, deren Erfüllung es durch das finale System zu beweisen gilt. Hierzu wird bevorzugt eine analytische Beweisführung genutzt. Ist dies nicht möglich, werden Testfälle abgeleitet, die zum Nachweis der Sicherheit zu erfüllen sind.

[31] Berg, G. et al.: Vehicle in the Loop (2016).

[32] Rüger, F. et al.: Kontrollierbarkeitsbewertung von FAS (2015).

2.2 Maschinelle Lernverfahren

Maschinelles Lernen bildet ein Teilgebiet der Künstlichen Intelligenz. Mit den Methoden des ML, den Lernalgorithmen, werden aus zur Verfügung gestellten Daten automatisiert Zusammenhänge extrahiert, die in Modellen gespeichert werden.[33] Diese Modelle werden genutzt, um Vorhersagen für neue, ungesehene Eingangsdaten zu treffen.[34] ML basiert daher auf dem Prinzip der Induktion.[35] Der Begriff ‚Lernen' wird verwendet, da es sich um den automatisierten Erwerb von Regeln oder die Verbesserung bestehender Regeln aus einer Datenmenge handelt.[36]

Im Folgenden wird allgemein auf den Entwicklungsprozess von gelernten Modellen eingegangen (Abschnitt 2.2.1). Hieran schließt sich eine Gliederung der zur Verfügung stehenden Lernverfahren in Abschnitt 2.2.2 an.

2.2.1 Entwicklungsprozess von maschinellem Lernen

Durch das induktive Vorgehen ist der Entwicklungsprozess von gelernten Modellen im Vergleich zu konventionell programmierten Modellen verändert. Während im Rahmen der konventionellen Programmierung zunächst die Ableitung von funktionalen und nichtfunktionalen Anforderungen vor Beginn der eigentlichen Modellbildung steht, ist dieser Schritt durch die automatisierte Generierung des Modells aus den zur Verfügung stehenden Daten nicht notwendig. Hierdurch entspricht der Entwicklungsprozess des maschinellen Lernens nicht dem anforderungsorientierten Vorgehen des V-Modells, welches in der ISO 26262 für die Systementwicklung gefordert ist (siehe Abschnitt 2.1.2).

Es existieren zwei grundsätzlich verschiedene Entwicklungsprozesse von ML, wobei lediglich einer der beiden für die zwei in der vorliegenden Arbeit fokussierten Algorithmenarten relevant ist. Die Unterscheidung der Algorithmenarten wird in Abschnitt 2.2.2 näher erläutert, die Fokussierung wird in Abschnitt 2.3.4 aus den aktuell in Fahrerassistenzsystemen verwendeten Algorithmenarten begründet.

Die Schritte des für die vorliegende Arbeit relevanten Entwicklungsprozesses sind im Folgenden dargestellt. Dabei existiert innerhalb der Vorgehensweise des Prozesses ab dem vierten Schritt eine Unterscheidung, ob eine Problemstellung vorliegt, bei dem die vorherzusagende Ausgangsgröße (sog. Label[37]) im Datensatz, der dem Algorithmus zur Verfü-

[33] Vgl. Awad, M.; Khanna, R.: Efficient Learning Machines (2015), S. 1.

[34] Vgl. Copeland, M.: Difference Between AI, Machine Learning, and Deep Learning (2016).

[35] Vgl. Bergadano, F.: The Problem of Induction and Machine Learning (1991), S. 1073.

[36] Vgl. Morik, K.: LS 8 Report 1, Maschinelles Lernen (1993), S. 2.

[37] Ein Label stellt eine Zielgröße dar, die mit jedem Objekt des Datensatzes verbunden ist. Vgl. Label (2017).

gung gestellt wird, vorhanden ist oder nicht. Die beiden Varianten des Entwicklungsprozesses sind in Abbildung 2-3 und Abbildung 2-4 dargestellt.

1. Datensammlung/ Datenvorauswahl: Aus der zur Verfügung stehenden Datenmenge wird eine Vorauswahl an den für die Problemstellung bzw. die zu lösende Aufgabe relevanten Daten getroffen.[38]
2. Vorverarbeitung der Daten: Je nach Ausgangslage der vorliegenden Daten ist eine Vorverarbeitung der Daten notwendig. Nach Abschluss der Vorverarbeitung ist folgender Stand der Daten zu erreichen:
    - Gemeinsames, dem Problem angemessenes Format der Daten
    - Frei von korrupten oder fehlerhaften Datenpunkten
    - Geeignete Abtastung der Daten[38]

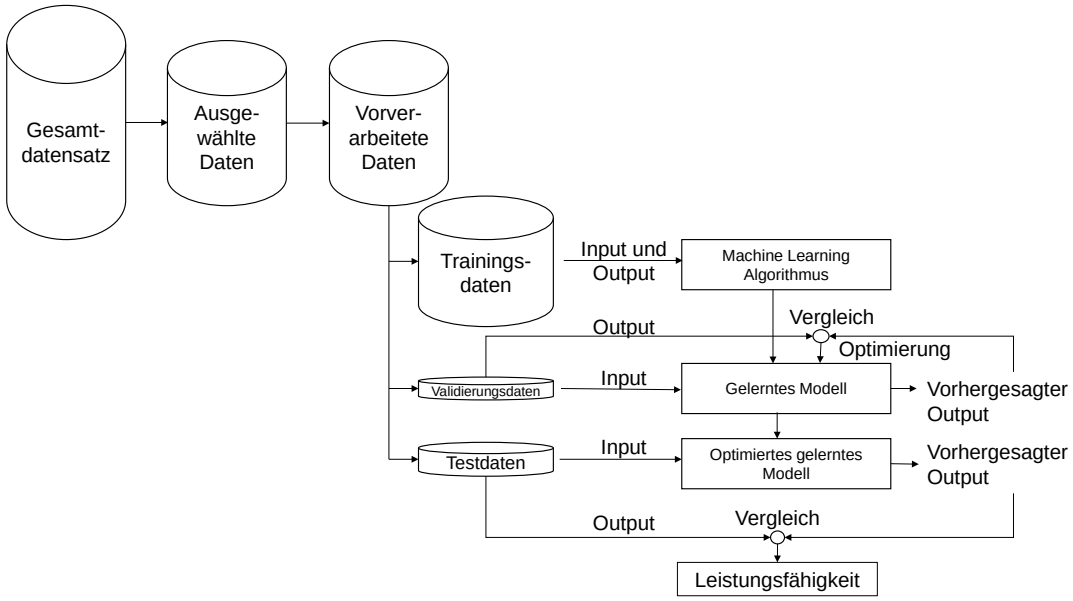


Abbildung 2-3: Entwicklungsprozess Algorithmus mit Labeln

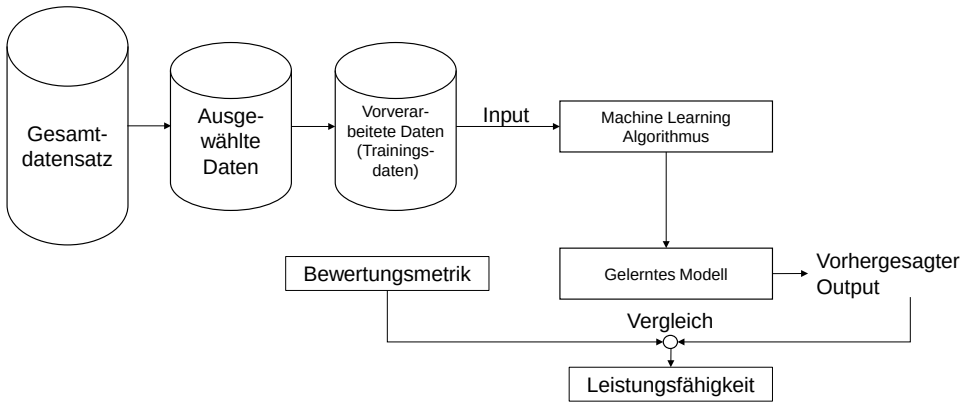


Abbildung 2-4: Entwicklungsprozess Algorithmus ohne Label

[38] Vgl. Brownlee, J.: How to Prepare Data For Machine Learning (2013).

3. Transformation der Daten: Der letzte Schritt in der Vorbereitung der Daten besteht in deren Transformation, um die zur Aufgabenerfüllung bzw. Problemlösung geeignete Datenstruktur, die die sogenannten Features beinhaltet, zu erhalten. Ein Feature im Sinne von ML ist eine Eigenschaft oder eine Variable, um einen Aspekt der individuellen Objekte, die in den Daten vorhanden sind, zu beschreiben.[39] Ein Beispiel hierfür sind beispielsweise Kraftstoffart sowie Außenfarbe bei Fahrzeugen. Im Rahmen der Transformation werden häufig folgende Operationen angewendet:

    - Skalierung: Viele Lernalgorithmen benötigen Daten, die auf den Wertebereich zwischen 0 und 1 skaliert sind.

    - Dekomposition: Einzelne Datenkanäle enthalten teilweise zur Problemlösung nicht-relevante Informationen. Ein Beispiel hierfür ist ein Datenkanal, welcher das Datum mit zugehöriger Uhrzeit enthält, wobei zur Problemlösung lediglich die Information, um welche Uhrzeit es sich in Stunden handelt, notwendig ist. Durch die Dekomposition werden solche komplexen Datenkanäle zerlegt und lediglich die relevanten Sub-Kanäle weiterverwendet.

    - Aggregation: Es ist möglich, dass das Zusammenführen einzelner Datenkanäle zu einem Kanal, welcher die Informationen durch domänenspezifisches Vorwissen miteinander kombiniert oder mathematische Operationen durchführt, hilfreich für die Problemlösung ist. Ein Beispiel hierfür ist die Verwendung der Beschleunigung als Ableitung der Geschwindigkeit, wenn das Signal der Beschleunigung nicht explizit vorliegt.[38]

4. Training und Auswahl des Algorithmus: Der transformierte Datensatz wird in einen Trainings- und Testdatensatz geteilt, wenn die gewünschten Ausgangsgrößen des Modells zur Verfügung stehen.[40] Der Trainingsdatensatz wird häufig zusätzlich noch in den tatsächlichen Trainingsteil und in einen Validierungsteil getrennt.[41] Wenn keine Ausgangsgrößen zur Verfügung stehen wird der komplette Datensatz zum Training genutzt.

    - Der Trainingsdatensatz wird dem Algorithmus mit Labeln (falls vorhanden) zur Verfügung gestellt. Aus diesem (Teil-)Datensatz werden durch den Algorithmus Zusammenhänge in den Daten extrahiert, was das eigentliche „Lernen“ bzw. „Training“ des Algorithmus darstellt.[40] [41] Hierzu wird das

[39] Vgl. Dong, G.; Liu, H.: Preliminaries and Overview (2018), S, 1f.

[40] Vgl. Awad, M.; Khanna, R.: Efficient Learning Machines (2015), S. 5f.

[41] Vgl. Shah, T.: About Train, Validation and Test Sets in Machine Learning (2017).

Optimum der „objective function“[42] gesucht, die den Kern des Lernalgorithmus darstellt. Die Optimierung basiert auf der Evaluation, wie falsch das gelernte Modell in Bezug auf seine Fähigkeit ist, die im Datensatz vorhandenen Beziehungen (bspw. zwischen Ein- und Ausgangsgrößen) abzuschätzen.[43]

- Bei Vorliegen von Annotationen (Labeln): Mit dem Validierungsdatensatz wird sich zwischen denen aus Training hervorgehenden Modellvarianten, die sich bspw. in ihren Hyperparametern[44] unterscheiden, entschieden, allerdings nicht weitertrainiert.[41]

In diesem Entwicklungsschritt findet ebenfalls die Auswahl eines konkreten Algorithmus anhand des Vergleichs der Leistungsfähigkeit statt, falls prinzipiell mehrere Algorithmen für die Problemstellung in Frage kommen.

5. Test des Modells: Mithilfe des Testdatensatzes wird die Leistungsfähigkeit des resultierenden gelernten Modells evaluiert. In diesem Schritt wird das Modell nicht weiter trainiert bzw. verändert.[40] [41] Wenn keine Label zur Verfügung stehen, wird die Leistungsfähigkeit in einer anderen Weise, wie z.B. durch die Auswertung der Trennbarkeit der unterschiedlichen Ausgangsgrößen, evaluiert.
6. Betrieb des Modells: Das gelernte Modell wird für seine Aufgabe eingesetzt.[40]

Es besteht die Möglichkeit, dass der Algorithmus die neuen, während seines Betriebs empfangenen Daten zum weiteren Training nutzt, wodurch das Modell stetig verbessert wird.[45] Bedingt hierdurch schließen sich weitere Schritte an den vorgestellten Entwicklungsprozess an. Diese Algorithmen werden als online-lernend bezeichnet.[45] Im Gegensatz hierzu stehen die offline-lernenden Modelle (auch Batch-Learning genannt), die sich während des Betriebs nicht verändern.[46] Dementsprechend ist deren Entwicklungsprozess mit dem sechsten Schritt abgeschlossen. Die möglichen weiteren Prozessschritte von Online-Learning werden im Folgenden aufgrund der aus Abschnitt 2.3.4 hervorgehenden Fokussierung der vorliegenden Arbeit nicht vorgestellt.

---

[42] Es wurde im Rahmen einer Literaturanalyse festgestellt, dass keine einheitliche Terminologie für die Benennung der Funktion eines Lernalgorithmus existiert. Neben „objective function“ findet sich beispielsweise der Begriff „cost function“, „loss function“, „error function“ (Vgl. Bishop, C. M.: Pattern recognition and machine learning (2006)). Im Folgenden wird der Begriff „objective function“ verwendet, da dieser sowohl eine mögliche Minimierung als auch eine Maximierung beinhaltet und einen etablierten Begriff aus der Mathematik darstellt (Vgl. Burke, J.: Linear Optimzation (2018)).

[43] Vgl. McDonald, C.: Machine learning fundamentals (I): Cost functions and gradient descent (2017).

[44] Hyperparameter, wie die Anzahl der zu identifizierenden Klassen, werden vor Beginn des Trainings festgelegt. Im Validierungsprozess wird bspw. aus einer Menge an Modellen mit unterschiedlichen Hyperparametern das Optimum ausgewählt.

[45] Vgl. Wachenfeld, W.; Winner, H.: Lernen autonome Fahrzeuge? (2015), S. 470.

[46] Vgl. Batch learning (2017).

2.2.2 Kategorien und Arten des Maschinellen Lernens

Maschinelle Lernalgorithmen lassen sich hinsichtlich verschiedener Kriterien gliedern. Eine weit verbreitete Möglichkeit besteht darin, die Algorithmen entsprechend der Art ihres Trainings bzw. ihres Lernens zu unterteilen, wobei diese Art von den zur Verfügung stehenden Informationen abhängt.[47] Die drei grundlegende Algorithmenkategorien entsprechend dieser Gliederung lauten:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning[48]

Bei **Supervised Learning** wird dem Algorithmus ein Datensatz mit Eingangs- und zugehöriger/ n Ausgangsgröße(n) bzw. Label(n) zum Training zur Verfügung gestellt. Hieraus extrahiert der Algorithmus Zusammenhänge zwischen den Eingangs- und Ausgangsgröße(n).[48] Die Ausgangsgröße(n) besitzen entweder einen kontinuierlichen Wertebereich (Regression) oder stellen einzelne Kategorien dar.[49] Mit dem hieraus resultierenden gelernten Modell werden für ungesehene Eingangsgrößen die entsprechenden Ausgangsgrößen vorhergesagt. Supervised Learning ist vergleichbar mit dem menschlichen Erlernen von Mustern: Bilder mit unterschiedlichen Verkehrszeichen werden Menschen gezeigt und die abgebildeten Verkehrszeichen durch eine externe Instanz benannt. Nach der Lernphase ist der Mensch in der Lage, diese Verkehrszeichen auch in anderen Bildern selbst zu identifizieren und korrekt zu benennen.[48]

Im Gegensatz zum Supervised Learning besitzt der dem Algorithmus zur Verfügung gestellte Datensatz beim **Unsupervised Learning** keine Label. In der Analogie zum Beispiel des menschlichen Lernens fehlt die Benennung der abgebildeten Formen durch eine externe Instanz. Durch das Fehlen von Label(n) ist es lediglich möglich, ähnliche Strukturen oder Gruppen innerhalb der Eingangsdaten zu identifizieren. Die hierzu gefundenen Regeln werden genutzt, um die Ähnlichkeit oder Gruppenzugehörigkeit von ungesehenen Eingangsdaten zum bestehenden Datensatz bzw. den hierin identifizierten Strukturen vorherzusagen.[48] Man unterscheidet in Clustering und Assoziation, wobei Clustering die Gliederung von Datenpunkten entsprechend ihrer Eingangsgrößen in einzelne Gruppen beinhaltet und Assoziation die Detektion von Regeln, die zwischen den einzelnen Datenpunkten vorherrschen.[50]

Im Rahmen von **Reinforcement Learning** beobachtet ein eingesetzter Algorithmus die aktuelle Umgebung und erhält ggf. speziell für ihn definierte Eingangsgrößen. Basierend

[47] Vgl. Maiß, C.: Masterthesis, Literatur- und Patentrecherche maschinelles Lernen (2016), S. 11.

[48] Vgl. Kwok, J. et al.: Machine Learning (2015), S. 496.

[49] Vgl. Ramasubramanian, K.; Singh, A.: Machine Learning Using R (2017), S. 1ff.

[50] Vgl. Morgun, I.: Types of machine learning algorithms (2015).

hierauf führt der Algorithmus eine Aktion aus und ändert hierdurch die Umgebung, wodurch er eine Bewertung (positives oder negatives Feedback) seiner durchgeführten Aktion erhält.[48] Der Algorithmus wird nicht, wie beim Supervised Learning, zur Auswahl einer bestimmten Aktion basierend auf speziellen Eingangsgrößen trainiert, sondern zur Entwicklung der bestmöglichen Handlungsstrategie durch Rückmeldung von Erfolg oder Misserfolg.[51] Erfolg oder Misserfolg ist dabei entweder direkt messbar (Klassifikation) oder lässt sich nicht direkt quantifizieren.[49] Nicht direkt quantifizierbar ist beispielsweise die Emotion des Fahrers, wenn eine Infotainment-Einstellung automatisiert geändert wird.

Diesen grundlegenden Kategorien lassen sich einzelne Algorithmenarten zuordnen, wobei die Algorithmenarten teilweise mehr als nur einer Kategorie zugehörig sind. Die einzelnen Algorithmenarten beschreiben Gruppen von Algorithmen, die das gleiche Funktionsprinzip besitzen. In der in Abbildung 2-5 dargestellten Zuordnung von Algorithmenarten zu den Algorithmenkategorien sind lediglich die Arten aufgeführt, die im Rahmen der vorliegenden Arbeit genannt werden. Am unteren Rand der Abbildung ist jeweils, falls möglich, ein Beispiel aus dem Bereich der Fahrzeugsysteme gegeben.

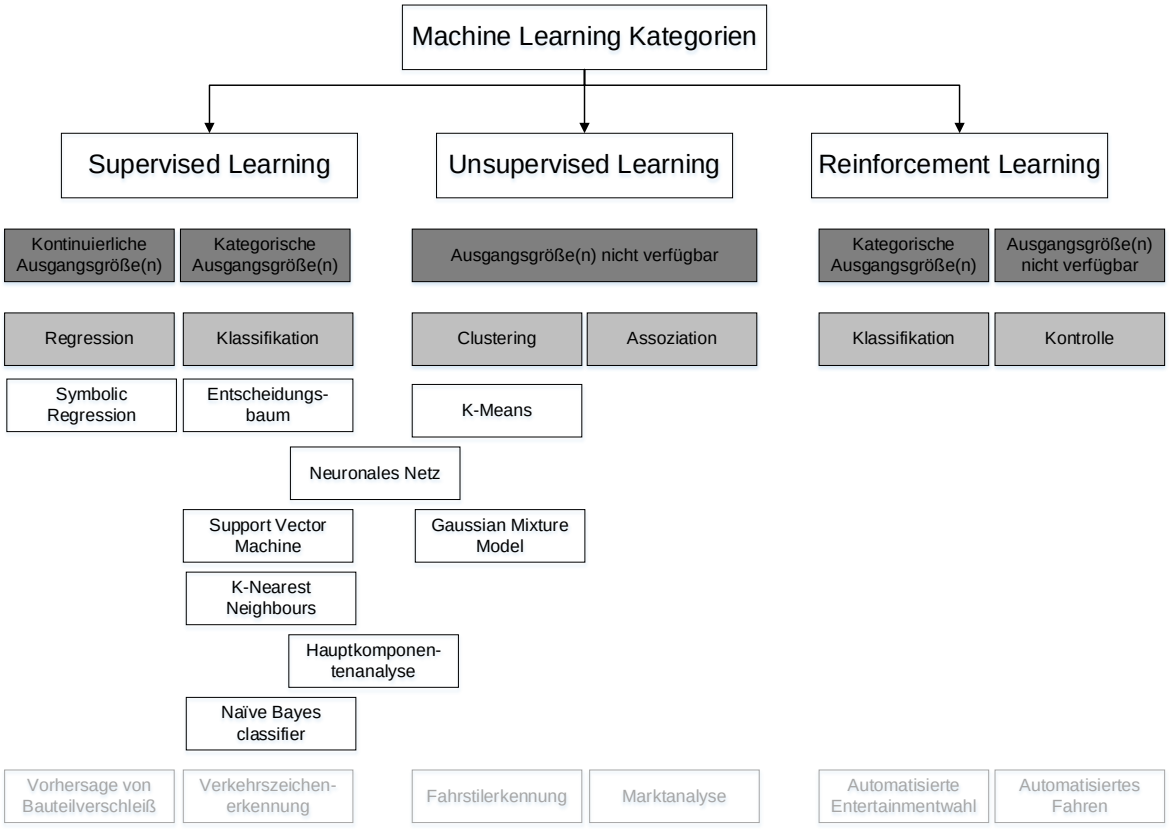


Abbildung 2-5: Zuordnung der ML-Arten zu ML-Kategorien nach Morgun[52]

Auf eine genauere Beschreibung aller aufgeführten Arten wird verzichtet, da die vorherrschende Verständnisebene zur Nachvollziehbarkeit der vorliegenden Arbeit ausreicht.[53]

[51] Vgl. Behera, R.; Das, K.: A Survey on Machine Learning (2017), S. 1304.

[52] Morgun, I.: Types of machine learning algorithms (2015).

Lediglich auf Neuronale Netze[54] (NN) wird im Folgenden näher eingegangen, da sich hierauf Erläuterungen in dem weiteren Verlauf der Arbeit beziehen. Zusätzlich wird der K-Means-Algorithmus im Rahmen des Abschnitts 6.2.1 vorgestellt.

Neuronale Netze sind eine Art der Algorithmen, deren Struktur an Gehirne angelehnt ist. Sie werden auch künstliche Neuronale Netze in Abgrenzung zu den natürlichen neuronalen Netzen, den Gehirnen von Lebewesen, genannt.[55] Da jedoch im Folgenden diese Abgrenzung nicht notwendig ist, wird der Begriff Neuronales Netz verwendet. Die Leistungsfähigkeit von NN ist im Vergleich zu einfachen Modellen, wie Entscheidungsbäumen, sehr hoch, weshalb sie häufig in komplexen Problemstellungen Anwendung finden. Jedoch zeichnen sie sich ebenfalls durch eine hohe Komplexität und einen hohen Rechenaufwand zum Training des Modells aus. Ein Neuronales Netz besteht aus einzelnen Neuronen, die jeweils in Schichten gruppiert sind (siehe Abbildung 2-6). Es existiert eine Eingabeschicht, eine oder mehrere verdeckte Schichten und eine Ausgabeschicht.[56] Die Neuronen sind miteinander durch Netzgewichte, dargestellt durch Pfeile, verbunden. Die Neuronen der Eingabeschicht repräsentieren die Eingangsgrößen, die dem Algorithmus zur Verfügung gestellt werden. Diese können eine Aktivierung der Neuronen in den verdeckten Schichten bewirken. Ob ein Neuron aktiviert wird, hängt vom eingehenden Wert, welcher sich aus dem Eingabewert und dem Netzgewicht berechnet, sowie einem Schwellwert mit dem der eingehende Wert verglichen wird, ab.[55] Dabei existieren verschiedene Ausführungsformen des Schwellwerts bzw. Aktivierungsfunktion in den Neuronen.[57] Die Neuronen der Ausgabeschicht bilden die geforderten Ausgangsgröße(n) ab.[55]

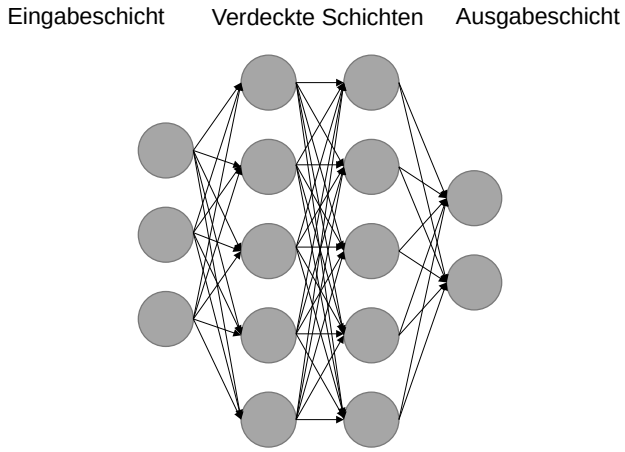


Abbildung 2-6: Neuronales Netz mit zwei verdeckten Schichten

[53] Bei Bedarf kann auf die Beschreibungen der Arten in Standardwerken des ML wie Mitchell, T. M.: Machine learning (1997). zurückgegriffen werden.

[54] Englisch: Neuronal Network.

[55] Vgl. Kramer, O.: Neuronale Netze (2009), S. 119ff.

[56] Vgl. Backhaus, K. et al.: Neuronale Netze (2018), S. 582.

[57] Vgl. V, A. S.: Understanding Activation Functions in Neural Networks (2017).

Abhängig vom Aufbau und der Struktur werden Neuronale Netze in einzelne Unterarten gegliedert, die sich für verschiedene Anwendungszwecke besonders gut eignen. Tiefe Neuronale Netze[58] zeichnen sich bspw. durch eine Anzahl von mindestens zwei verdeckten Schichten aus.[59] Die Spezialform des Convolutional Neuronal Networks (CNN) eignet sich beispielsweise besonders zur Verarbeitung von Bild- und Audiodaten. Ein CNN besitzt hierfür eine spezielle Abfolge von Schichten, die u.a. Faltungsoperationen[60] durchführen.[61]

2.3 Anwendungsbereiche von maschinellen Lernverfahren in FAS

Ausgehend von einer Literatur- und Patentrecherche für Anwendungen von maschinellen Lernalgorithmen im Fahrzeug nach Maiß[62] werden im Folgenden die hierin identifizierten Anwendungen für Fahrerassistenzsysteme zusammengefasst und durch eine erweiterte Literaturrecherche ergänzt. Die Arbeit von Maiß befasst sich grundsätzlich mit den Anwendungen des Maschinellen Lernens in FAS und gliedert diese in die Anwendungsfelder Sicherheit, Komfort und Effizienz. Diese Kategorisierung wird jedoch nicht übernommen, da die Gliederung der Anwendungsfelder anhand des Einsatzes in der Wirkungskette von Fahrerassistenzsystemen „Sense-Plan-Act“, welche aus der Robotik bekannt ist,[63] für die Verständlichkeit der vorliegenden Arbeit besser geeignet ist. Daher werden aus der Recherche lediglich die identifizierten Anwendungen von ML übernommen und in einen neuen Kontext eingeordnet. Wie bereits erwähnt werden die durch Maiß[62] identifizierten Anwendungen darüber hinaus noch erweitert.

Aus dieser neuen Gliederung resultieren die Kategorien „Wahrnehmung“, „Planung“ und „Aktion“. Gesamtsysteme, in denen sich der Einsatz der Lernalgorithmen nicht nur auf eine dieser Kategorien beschränkt, werden entsprechend ihrer Ausgangsgröße kategorisiert. Besteht die Ausgangsgröße beispielsweise in einer Beschreibung von Fahrereigenschaften wie der Fahrermüdigkeit, wird sie dem Bereich „Sensorik“ zugeordnet, da lediglich eine Verarbeitung von Signalen in eine Zustandsgröße stattfindet, die von anderen Funktionen, wie einer Pausenempfehlung, genutzt wird. Ist die Ausgangsgröße des Algorithmus z.B. eine direkte Beschleunigungsvorgabe an das Aktorsubsystem wird diese Anwendung im Bereich „Aktion“ vorgestellt, auch wenn der Algorithmus diese auf Basis

[58] Englisch: Deep Neuronal Networks

[59] Vgl. Tch, A.: The mostly complete chart of Neural Networks, explained (2017).

[60] Durch die Faltungsoperationen (engl. convolutional) ergibt sich die Benennung dieser Spezialform.

[61] Vgl. Lubner, S.; Litzel, N.: Was ist ein Convolutional Neural Network? (2019).

[62] Maiß, C.: Masterthesis, Literatur- und Patentrecherche maschinelles Lernen (2016).

[63] Vgl. Kortenkamp, D.; Simmons, R.: Robotic Systems Architectures and Programming (2008), S. 189.

von Bilddaten berechnet und plant, d.h. die die Bereiche „Wahrnehmung“ und „Planung“ ebenfalls beinhaltet. Der Abschluss dieses Unterkapitels bildet eine Zusammenfassung, bei denen die identifizierten Anwendungsfelder umrissen und die derzeit am häufigsten Anwendung findenden Algorithmenarten identifiziert werden.

2.3.1 Wahrnehmung

Sensorik im Fahrzeug dient zur Aufnahme der realen Fahrzeuginnen- und –außenwelt zur weiteren Verarbeitung für Fahrzeugfunktionen. Hierzu gehören Ego-Fahrzeuggrößen, Größen der Umwelt sowie Merkmale des Fahrers, wie z.B. dessen Aufmerksamkeit. Maschinelles Lernen wird hier im Bereich des Rechnersehens bzw. Computer-Vision sowie der Verarbeitung und Klassifizierung von anderen Sensorsignalen für Fahrerassistenzsysteme wie Radar und Lidar genutzt. Aber auch im Bereich der Sprachsteuerung im Fahrzeug finden sich Anwendungen von ML. Neben der Verarbeitung von einzelnen Sensorsignalen existieren weitere Anwendungen zur Weiterverarbeitung bzw. Fusion verschiedener Merkmale, damit beispielsweise aus einzelnen messbaren Größen des Fahrers dessen Aufmerksamkeit abgeschätzt wird.[64] Die nachfolgende Beschreibung gliedert sich daher einerseits in die Verarbeitung der im Fahrzeug direkt aufgenommenen Sensorsignale wie Bilddateien oder Radarsignale und anderseits in die Modellierung von weiteren Zustandsgrößen, die nicht direkt messbar oder einfach aus einer einzigen Signalart ableitbar sind, wie beispielsweise die Fahrerintention.

2.3.1.1 Sensorsignale

Im Bereich der Fahrerassistenz werden neben Kameras vor allem Radar-, Lidar- und Ultraschallsensoren zur Umwelterfassung eingesetzt.[65] Auch der Innenraum wird im Rahmen von Fahrerassistenzsystemen erfasst, wobei hier Kameras, aber beispielsweise auch Mikrophone zum Einsatz kommen. Die Analyse der Anwendungsbereiche von Lernalgorithmen im Rahmen der Sensorsignale wird sich an der Verarbeitung von Bilddateien orientieren, da dies durch die Thematik der Computer-Vision bereits ein strukturiertes Forschungsgebiet darstellt.[66] Neben dem jeweiligen Anwendungsbeispiel aus der Bildverarbeitung wird eines aus einer anderen Sensorquelle genutzt, um die Übertragbarkeit darzustellen.

Bilddaten dienen in Fahrerassistenzsystemen zur Erfassung der Umwelt sowie des Fahrers bzw. weiterer Insassen, wobei diese Anwendung dem Bereich der Computer-Vision zuzuschreiben ist. Computer-Vision lässt sich in Bildaufnahme und Speicherung, Bildverarbei-

[64] Vgl. Batista, J. P.: A Real-Time Driver Visual Attention Monitoring System (2005).

[65] Vgl. Winner, H.: Handbuch Fahrerassistenzsysteme (2015), S. 221.

[66] Vgl. Priese, L.: Computer Vision (2015), 1f.

tung und Bildanalyse untergliedern, wobei die Bereiche Bildverarbeitung und –analyse ineinander übergehen.

Im Suchraum zur Kalibrierung von Kameras wurde ein Patent zur Verminderung des Einflusses von Nebel mit Support Vector Machines (SVM)[67] identifiziert, wobei mit diesem ein Randbereich der Anwendungen von Maschinellem Lernen in Computer-Vision adressiert ist. Im Bereich der Kalibrierung anderer Sensoren wurde keine Anwendung identifiziert.

Der Bildanalyse werden dabei die Aufbereitungsschritte

- Segmentierung und
- Identifikation von Objekten sowie
- Analyse elementarer Formen

zugeschrieben.[66] Da es möglich ist, die Bildanalyse als Teilbereich der Künstlichen Intelligenz zu verstehen, findet hierin Maschinelles Lernen Anwendung, um die bisherigen menschlich programmierten Umwandlungen der Bilder in interne Repräsentationen zu verbessern. [68] Neben den Bereichen der klassischen Bildanalyse wird ML im Bereich der Computer-Vision genutzt, um interne Repräsentationen so zu verwandeln, damit sie dem benötigten Wissen zur Aufgabenerfüllung entsprechen.[68]

Im Rahmen der Segmentierung gilt es, zusammengehörige Bildbereiche einander auf Pixelebene zuzuordnen, jedoch ohne sie weiter zu klassifizieren.[69] Hierdurch ist es möglich, verschiedene zusammenhängenden Flächen von Umweltaufnahmen, wie bspw. eine Straße, ein Fahrzeug oder Randbegründung, voneinander zu unterscheiden oder zu identifizieren. Es ist möglich, sowohl Supervised- als auch Unsupervised-Ansätze zu verwenden. Eine detaillierte domänenübergreifende Zusammenfassung der Algorithmenansätze findet sich bei Thoma[69]. Beispiele für die Segmentierung aus dem Bereich der Fahrerassistenz stellen die Segmentierung von Lichtsignalanlagen[70], in welchem zwei Klassifizierer eingesetzt werden, sowie die Segmentierung von Straßen[71] dar. Segmentierung findet ebenfalls im Rahmen von anderen Sensorquellen wie Radarsensorik statt. Hierbei werden die zugehörigen Bereiche in einer alternativen Weltdarstellung, wie einem sogenannten Occupancy Grid, einander zugeordnet. Dies wird beispielsweise verwendet, um Bereiche mit parkenden Fahrzeugen mit einem Convolutional Neuronal Network von anderen Bereichen zu trennen, wodurch implizit gleichzeitig eine Klassifikation erfolgt.[72]

---

[67] Vgl. Stein, G.: Bundling of driver assistance systems (2010).

[68] Vgl. Sebe, N. et al.: Machine Learning in Computer Vision (2005), S. 3.

[69] Vgl. Thoma, M.: A survey of semantic segmentation (2016), S. 1.

[70] Vgl. Haltakov, V. et al.: Semantic Segmentation (2015).

[71] Vgl. Kuhnl, T. et al.: Monocular road segmentation using slow feature analysis (2011).

[72] Vgl. Lombacher, J. et al.: Semantic radar grids (2017).

Die Identifikation von Objekten erfolgt mit maschinellen Lernverfahren auf zwei Weisen. Einerseits ist es möglich, Objekte direkt anhand der Rohbilddateien, d.h. auf Pixelbasis, zu detektieren und klassifizieren. Andererseits werden Lernalgorithmen auch auf bereits vorverarbeitete Bildmerkmale, die bereits Objekte detektieren, angewendet, um hierdurch eine Klassifikation vorzunehmen. Die vorverarbeiteten Bildmerkmale sind dabei beispielsweise Kanten oder Intensitätsunterschiede in Bildzellen, auf denen der Lernalgorithmus trainiert wird. Dieser entscheidet dann z.B. anhand von gerichteten Gradientenvektoren, ob das Objekt ein Fußgänger ist oder nicht.[73] Auch im Rahmen der Innenraumbeobachtung wird die Objektklassifikation mittels maschineller Lernverfahren genutzt, um beispielsweise die Blickrichtung des Fahrers anzugeben. Dabei wird ein online-lernender Ansatz verwendet, der das Blickmodell über der Betrachtungsdauer des Fahrers weiter an das Individuum anpasst.[74] Durch die generelle Verwendung von vorverarbeiteten Merkmalen und der hiermit verbundenen Nutzung von Ad-hoc-Domänenwissen[75] ist es möglich, weniger komplexe und rechenintensive Lernalgorithmen zur Klassifikation auf Merkmalsbasis einzusetzen als bei einer Objektklassifikation auf Pixelbasis. Eingesetzt werden aus dem Bereich der Supervised-Learning-Algorithmen unter anderem Support-Vector-Machines, k-Nearest-Neighbours und Entscheidungsbäume (Decision Trees). Aus dem Bereich der Unsupervised-Algorithmen finden beispielsweise K-Means-Clustering und Gaussian-Mixture-Models (GMM) Anwendung.[76] Ist genügend Rechenleistung vorhanden, wie eine Rechenplattform des Herstellers NVIDIA[77], ist es beispielsweise möglich, ein CNN für die Verkehrszeichenerkennung einzusetzen, um direkt auf Pixelbasis Objekte zu klassifizieren.[78] Bei der Nutzung anderer, nicht kamerabasierter Sensordaten, ist eine Objektklassifizierung ebenfalls auf Rohdatenbasis oder hieraus extrahierten Merkmalen möglich. Bei der Benutzung von Radarsensoren ist besteht eine Anwendung darin, auf Basis des Occupancy Grids, entweder direkt oder über vorverarbeitete Merkmale die Art der Straße zu klassifizieren. Hierbei wird für die direkt aus den Rohdaten vorgenommene Klassifizierung ein Neuronales Netz und für die auf Merkmalen basierte Klassifizierung eine Support-Vektor-Machine genutzt, wobei die Leistung der SVM in diesem Anwendungsfall ähnlich hoch der des NN ist. Dies ist mit dem relativ kleinen Trainingsdatensatz begründet, da eigentlich erwartet wurde, dass das NN eine deutlich höhere Klassifizierungsleistung besitzt.[79] Ein weiteres Beispiel auf Basis von vorverarbeiteten Merkmalen stellt die Identifikation von Fußgängern aus Lidar-Punktewolken dar. Hierbei fand ein Vergleich von drei verschiede-

[73] Vgl. Rezaei, M.; Klette, R.: Computer Vision for Driver Assistance (2017), 59ff.

[74] Vgl. Smart Eye AB: Technology | Smart Eye (2016).

[75] Vgl. Viola, P.; Jones, M. J.: Robust Real-Time Face Detection (2004), S. 139.

[76] Vgl. Rezaei, M.; Klette, R.: Computer Vision for Driver Assistance (2017), S. 53.

[77] Vgl. Said, C.: Driving the future (2017).

[78] Vgl. Filkovic, I.: Traffic Sign Localization and Classification Methods: An Overview (2014), 5ff.

[79] Vgl. Seeger, C. et al.: Towards road type classification with occupancy grids (2016).

nen Lernalgorithmen, k-Nearest-Neighbours, Naïve-Bayes-Classifier und Support-Vector-Machine statt, wobei die SVM die besten Ergebnisse erzielte.[80]

Die Analyse elementarer Formen findet sich im Bereich von Kameradaten im Rahmen der Ego-Lokalisierung wieder. Zur Lokalisierung werden Formen der Umwelt aufgezeichnet und inkl. deren (Referenz-)Position gespeichert, um sich bei Wiedererkennen der Formen auf deren Position zu beziehen. Es besteht die Möglichkeit Lernen zur Extraktion von geeigneten Lokalisierungsmerkmalen, in diesem Anwendungsbereich Landmarken genannt, einzusetzen[81] oder Rohbilder anhand ihrer Ähnlichkeit ohne explizite Extraktion von Merkmalen einer bestehenden Bildbasis zuzuordnen.[82] Hierzu werden unter anderem CNNs genutzt.[81] [83] Auch im Bereich von anderen Sensorsignalen wird maschinelles Lernen zur Analyse elementarer Formen zur weiteren Verarbeitung genutzt. Hierbei werden beispielsweise aus 3-D-Punktewolken eines Lidar-Sensors Formen wie horizontale oder vertikale Ausbreitungen extrahiert, um sie später zur Segmentierung oder zur Objektklassifizierung zu nutzen. Für diese Analyse wurden verschiedene Supervised- Algorithmen, wie SVM und GMM, analysiert, wobei mit Neuronalen Netzen die höchste Performance erreicht wurde.[84] Ein weiteres Beispiel für die Analyse elementarer Formen aus dem Bereich der Innenraumerkennung stellt die Spracherkennung dar, bei der die akustischen Signale in Merkmalsvektoren konvertiert und anschließend zu Wörtern bzw. Sätzen dekodiert werden. Aufgrund der Abhängigkeiten von Wörtern innerhalb eines Satzes, eignen sich probabilistische Algorithmen, aber auch NN, zum Erlernen der Sprache.[85] BMW brachte diese Anwendung maschinellen Lernens 2015 in Serie.[86]

2.3.1.2 Weitere Zustandsgrößen

Neben den Anwendungen im Bereich der Rohsignalverarbeitung wird maschinelles Lernen in Fahrerassistenzsystemen dazu verwendet, mehrere unterschiedliche Rohsignale miteinander zu kombinieren, um hieraus Zustandsgrößen zu erhalten, die nicht oder nur mit geringerer Qualität aus einer einzigen Sensorquelle ableitbar sind. Ein Beispiel hierfür stellt die Abschätzung der kognitiven Fahrerbelastung dar. Auf Basis von Blickbewegungen, welche aus Kamerabildern extrahiert werden, und anderen Merkmalen, wie beispielsweise der Lenkwinkel und der Fahrzeugbeschleunigung, wird ein Zusammenhang zur aktuellen Belastungen des Fahrers gelernt. Hierzu wird z.B. ein Entscheidungsbaum genutzt,

[80] Vgl. Navarro, P. J. et al.: Pedestrian Detection for Autonomous Vehicles (2016).

[81] Vgl. Arroyo, R. et al.: Fusion and binarization of CNN features (2016).

[82] Vgl. Walch, F.: Masterthesis, Deep Learning for Image-Based Localization (2016), S. 23.

[83] Vgl. Thrun, S.: Bayesian Landmark Learning (1998).

[84] Vgl. Plaza-Leiva, V. et al.: Classification of Lidar Point Clouds (2017).

[85] Vgl. Fakotakis, N.; Sgarbas, K. N.: Machine Learning in Human Language Technology (2001), S. 668.

[86] Vgl. Eddy, N.: Machine Learning Drive (2016).

wenn gelabelte Belastungen des Fahrers vorliegen. Auf Basis der hierdurch ermittelten Fahrerbelastung ist es möglich, die Interaktion mit dem Fahrzeug über die Mensch-Maschine-Schnittstelle an den aktuellen Belastungszustand anzupassen.[87] Auch zur Abschätzung von Manöverintentionen, wie einer Überhol-Intention, wurde maschinelles Lernen bereits eingesetzt.[88] Es existieren ganze Forschungsgruppen, die sich mit der Vorhersage von Fahrerverhalten bzw. deren Manöverintentionen mittels Neuronaler Netze beschäftigen und, laut eigenen Angaben, einen Vorhersagehorizont von zu 3,5 s erreichen.[89] Neben fahrerzentrierten Zustandsgrößen ist es auch möglich, die Kritikalität der aktuellen Verkehrssituation anhand verschiedener Merkmale zu erlernen, um hieran die Ausgabemodalität der Mensch-Maschine-Schnittstelle anzupassen.[90] Auch die Vorverarbeitung von Situationen zur Bewertung der Notwendigkeit für Fahrmanöver auf Basis von Bilddateien mit einem NN[91] stellt eine solche Zustandsgröße dar.

Für das Erlernen von Zustandsgrößen aus kombinierten Einzelsignalen werden eine Vielzahl an unterschiedlichen Algorithmen eingesetzt. Angewendet wurden beispielsweise bereits SVM und probabilistische Modelle.[92] Je nach Anzahl der kombinierten Signale sowie der Komplexität der Zustandsgröße ist es sinnvoll, NN zu verwenden.[93]

2.3.2 Planung

Der Bereich „Planung“ umfasst alle Anwendungen, deren Ausgangsgröße(n) eine Anweisung zur Rückmeldung an die Schnittstelle zum Fahrer bzw. an die Aktoren des Fahrzeugs enthalten, ohne dass diese Anweisung bereits ausgeführt wurde. Forschungen in diesem Bereich gliedern sich in zwei Kategorien. Einerseits wird Maschinelles Lernen zur Individualisierung von Systemen bzw. deren Planung eingesetzt, andererseits zur nicht-individuellen Generierung von Planungsmodellen.

Die Systemindividualisierung erfolgt auf verschiedene Weisen. Es ist möglich, dass sich die Planung an das spezifische Fahrzeug, den spezifischen Fahrer oder die spezifische gefahrene Umwelt (bspw. Gewohnheitsstrecken) anpasst. Dabei finden sich zwar Anwendungen bzw. Patente zur individuellen Anpassung an das Fahrzeug, wie ein bauteilindividuelles Erlernen von Parametern für Synchronisationsschwellen eines Doppelkupplungsge-

[87] Vgl. Zhang, Y. et al.: Learning-Based Driver Workload Estimation (2008), 5ff.

[88] Vgl. Dokania, P. et al.: Online lane change intention prediction (2013).

[89] Vgl. Brain4Cars (2016).

[90] Vgl. Bouzouraa, M. E.: Verfahren zum Betreiben einer Mensch-Maschine-Schnittstelle (2014).

[91] Vgl. Chen, C. et al.: DeepDriving (2015).

[92] Vgl. Mandalia, H. M.; Salvucci, M. D.: Using Support Vector Machines for Lane-Change (2016).

[93] Vgl. Maiß, C.: Masterthesis, Literatur- und Patentrecherche maschinelles Lernen (2016), 46 f.

triebes[94], allerdings sind diese Systeme nicht dem Bereich der Fahrerassistenzsysteme zuzuordnen. Ein Beispiel für die Anpassung auf den spezifischen Fahrer findet sich im Rahmen der Individualisierung von Adaptive Cruise Control (ACC) in der Forschung wieder. In einer Forschungsarbeit werden Regressions-Algorithmen und Entscheidungsbäume genutzt, um die einzuregelnde Zeitlücke zu individualisieren. Zur Anpassung werden Merkmale des gezeigten Fahrverhaltens, aber auch bekannte demographischen Informationen des Fahrers, genutzt.[95] Daneben existieren Ansätze, die nicht nur einzelne Parameter der ACC-Funktionalität, wie die Zeitlücke, zur Fahrerindividualisierung nutzen, sondern das gesamte Planungsmodul zur Beschleunigungsvorgabe fahrerspezifisch mittels Reinforcement-Learning generieren.[96] An der fahrerspezifischen Individualisierung wird auch im Rahmen von vollautomatisierten Überholvorgängen auf Autobahnen geforscht. Auch in diesem Fall werden die Parameter des Fahrstreifenwechsels, wie die akzeptierte Zeitlücke beim Ausscheren, an die gezeigte Fahrweise durch die Verwendung eines Gaussian Mixture-Models angepasst.[97a] Durch eine solche fahrerspezifische Individualisierung wird eine erhöhte Nutzerakzeptanz erwartet.[97b] Als Beispiel für eine umweltspezifische Individualisierung durch Maschinelles Lernen dient ebenfalls die ACC-Zeitlücke, welche situationsspezifisch erlernt wird, um beispielsweise ungewolltes Einscheren vor dem Ego-Fahrzeug bei Nutzung der ACC-Funktion auf Autobahnen zu vermeiden.[98]

Im Bereich der Individualisierung findet eine große Bandbreite von maschinellen Algorithmen Anwendung. Es werden sowohl Reinforcement-Learning-Ansätze als auch Entscheidungsbäume oder GMM eingesetzt. Die eingesetzten Algorithmen hängen stark von der Komplexität der Individualisierung ab. Passt sich ein System beispielsweise lediglich an die gezeigten Vorlieben eines Nutzers an (bspw. der Nutzung von Infotainmentkanälen), ist es möglich, einen weniger komplexen Algorithmus einzusetzen. Werden allerdings weitere Umfeldinformationen und die Reaktion des Fahrers auf diese mit in die Betrachtung einbezogen, eignen sich komplexere Algorithmen wie NN. Die eingesetzten Algorithmen unterscheiden sich in diesem Anwendungsfeld zudem darin, ob die Individualisierung während der Fahrt, d.h. online, oder ob diese in Betriebspausen, also offline, stattfindet.

Das Erlernen von generischen Planungsmodellen erfolgt in unterschiedlichen Funktionsumfängen. Durch den Einsatz von Maschinellem Lernen wird in diesem Anwendungsbereich die händische Programmierung durch die automatisierte Generierung von Modellen zur Planung ersetzt. Die Anwendungen reichen von einem situationsbasierten Motor-

[94] Vgl. Schmitt, W.: Verfahren zum Lernen von Synchronisationsschwellen (2016).

[95] Vgl. Rosenfeld, A. et al.: Improve the Acceptance of ACC (2012).

[96] Vgl. Chen, X. et al.: A learning model for personalized adaptive cruise control (2017).

[97] Vgl. Butakov, V.; Ioannou, P.: Personalized Driver/Vehicle Lane Change (2015). a: S. 4423ff.; b: S. 4430.

[98] Vgl. Rebhan, S.; Kleinehagenbrock, M.: Intelligent gap setting (2016).

management bei Hybridfahrzeugen mittels NN[99] bis hin zum Einsatz von Reinforcement-Learning für die Auswahl von definierten Fahrmanövern auf Autobahnen[100]. Ein anderer Ansatz besteht darin, Reinforcement-Learning zur Planung der einzuregelnden Zeitlücke für ein ACC-System einzusetzen, wobei der Abstand sowie dessen Ableitung zum Vorderfahrzeug als Eingangsmerkmale dienen.[101] Ein weiteres Beispiel in dieser Kategorie bildet ALVINN, ein Fahrzeug, dessen Planungsmodul mittels eines Neuronalen Netzes automatisiert eine Krümmung ausgibt, um der, vor dem Fahrzeug befindlichen, Straße zu folgen. Die Eingangsdaten des Netzes zur Krümmungsberechnung sind dabei Bilder der Straße sowie eine Abstandsmessung.[102] Bedingt durch die hohe Komplexität sowie die hohe Anzahl an Eingangsmerkmalen der Planungsmodelle werden häufig hochdimensionale, komplexe Algorithmen wie NN verwendet.

2.3.3 Aktion

„Aktion“ beschreibt die Anwendungen, die den Aktoren des Fahrzeugs eine direkte einzuregelnde Größe vorgeben. Eine Anwendung aus diesem Feld benutzt beispielsweise Neuronale Netze, um eine automatisierte Fahrfunktion zu modellieren. Durch das NN werden zugehörig zu Rohbilddateien direkte Steuergrößen zu Beschleunigung und Trajektorienverlauf ausgegeben.[103] Ansätze, die diesen breiten Funktionsumfang besitzen, werden als „end-to-end“-Learning bezeichnet werden, da sie ausgehend von Rohdaten eine Aufgabe wie die Ausgabe der Lenkungsansteuerung vornehmen.[104] Zu dem Anwendungsbereich „Aktion“ lassen sich jedoch wenige andere Anwendungen finden, da die Anwendungen häufig einen konventionellen Regler zwischen dem maschinell gelernten Modell und den Aktoren nutzen, wodurch sie zum Bereich „Planung“ gezählt werden.

2.3.4 Zusammenfassung

In Fahrerassistenzsystemen finden sich derzeit vor allem Anwendungen von maschinellem Lernen im Bereich der Sensorrohdatenverarbeitung, Modellierung von Zwischengrößen sowie Individualisierung von Planungsmodulen. In den ersten beiden Anwendungsbereichen liegt dies unter anderem darin begründet, dass es sich um hochkomplexe Problemstellungen handelt, bei denen die traditionelle Programmierung beispielsweise durch die Begrenztheit des menschlichen Vorstellungsvermögens auf wenige Dimensionen die

[99] Vgl. Park, J. et al.: Intelligent Energy Management and Optimization (2016).

[100] Vgl. Mirchevska, B. et al.: Reinforcement Learning for Autonomous Maneuvering (2017).

[101] Vgl. Desjardins, C.; Chaib-draa, B.: Cooperative Adaptive Cruise Control (2011), S. 1257.

[102] Vgl. Pomerleau, D. A.: ALVINN (1989), S. 306.

[103] Vgl. Bojarski, M. et al.: End to end learning for self-driving cars (2016).

[104] Vgl. Di, W. et al.: Deep Learning Essentials (2018), S. 33.

Leistungsfähigkeit von hochdimensionalen maschinell gelernten Modellen nicht mehr erreicht.[105] Es wird sogar die generelle Detektionsleistung von Menschen z.B. bei der Erkennung Verkehrszeichen von maschinell gelernten Algorithmen überboten.[106] Zusätzlich zeichnet sich ML dadurch aus, dass es keine vollständige funktionale Spezifikation im Vorfeld benötigt, da das zur Problemlösung benötigte Wissen implizit durch den Trainingsdatensatz gegeben wird. Daher wird maschinelles Lernen vor allem dann eingesetzt, wenn Probleme bzw. Aufgaben, wie im Fall der Objekterkennung, nicht vollständig spezifizierbar sind.[107] [108] Im Rahmen der Individualisierung lässt sich dies dadurch erklären, dass eine komplexe, stufenlos variable Modellanpassung an Fahrer, Fahrzeug oder Umwelt während des Betriebs lediglich mittels maschinellen Lernen umsetzbar ist. Die weniger komplexe Anpassung von einzelnen Parametern, beispielsweise anhand des gezeigten Fahrstils, lässt sich hingegen ebenfalls mittels anderer Algorithmen verwirklichen. Hierzu sind zum Beispiel offline verschiedene Fahrertypen zu definieren (bspw. mittels traditioneller Programmierung), die während der Fahrt erkannt und auf die die Parameteranpassungen vorgenommen werden.

Zur Umsetzung der vorgestellten derzeitigen Haupt-Anwendungsbereiche von Maschinellem Lernen in FAS werden eine Vielzahl an unterschiedlichen Algorithmen genutzt. Allein elf unterschiedliche Algorithmen wurden in obigen Abschnitten als Beispiele für die Anwendungsbereiche aufgelistet, wobei im Rahmen der Auswahl der einzelnen Beispiele kein Fokus auf eine möglichst hohe Variabilität der vorgestellten Ansätze gelegt wurde. Die Vielzahl an unterschiedlichen eingesetzten Algorithmen wurde ebenfalls von Maiß[109] festgestellt. Die hohe Variabilität lässt sich unter anderem mit der hohen Vielfalt an Anwendungsmöglichkeiten innerhalb von Fahrerassistenzfunktionen erklären, da abhängig von der zu lösenden Aufgabe ein anderer Algorithmus bzw. Algorithmentyp zu nutzen ist. Am mit Abstand häufigsten fand sich innerhalb der Literaturrecherche die Verwendung Neuronaler Netze und deren Spezialfälle wie CNN wieder. In der vorgestellten Sichtprobenmenge sind ebenfalls SVM, GMM und Entscheidungsbäume stark vertreten. Der häufige Einsatz von NN und SVM wird ebenfalls von Maiß[109] festgestellt. In Bezug auf die Lerntypen (siehe Abschnitt 2.2.2) finden sich daher hauptsächlich Supervised- und Unsupervised-Ansätze wieder, die offline gelernt wurden. Auch die Durchführung einer expliziten Literaturrecherche nach Anwendungen des Reinforcement- und Online-Learning bestätigt diese Aussage.

Die Forschungsfrage *„Welche ML-Arten werden in welchen Bereichen bei FAS eingesetzt?“* (vgl. Unterkapitel 1.2) lässt sich daher wie folgt beantworten:

---

[105] Vgl. Olah, C.: Visualizing MNIST: An Exploration of Dimensionality Reduction (2014).

[106] Vgl. Stallkamp, J. et al.: Man vs. computer (2012).

[107] Vgl. Spanfelner, B. et al.: Challenges in applying the ISO 26262 (2012), S. 10f.

[108] Vgl. Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018), S. 7.

[109] Maiß, C.: Masterthesis, Literatur- und Patentrecherche maschinelles Lernen (2016).

Es werden vor allem offline gelernte Supervised- und Unsupervised-Ansätze in den Bereichen Sensorrohdatenverarbeitung, Modellierung von Zwischengrößen sowie Individualisierung von Planungsmodulen eingesetzt.

3 Ist-Analyse des Sicherheitsnachweises von ML

Ausgehend von der Recherche der eingesetzten Arten von maschinellem Lernen in Fahrerassistenzsystemen und den dahinterstehenden Unterschieden im Entwicklungsprozess der gelernten Algorithmen (siehe Abschnitt 2.2.1) im Vergleich zu konventionell programmierten Modellen wird festgestellt, dass das aktuelle, durch die ISO 26262[110], vorgeschriebene Vorgehen zum Nachweis der Sicherheit von Fahrerassistenzsystemen (siehe Unterkapitel 2.1) nicht anwendbar ist. Die Richtlinien der ISO/ PAS 21448[111] stehen hingegen im Einklang mit der Verwendung von ML bzw. berücksichtigen diese Art der Modellierung explizit und geben Hinweise zur Qualifizierung von gelernten Modellen, was in Unterkapitel 3.1 vorgestellt wird. Jedoch lässt sich aus der ISO/ PAS 21448 kein eigenständiger Sicherheitsnachweis ableiten, weshalb die Widersprüche zwischen den eingesetzten Algorithmenarten und der ISO 26262 in Unterkapitel 3.2 vorgestellt werden, um hierauf basierend bestehende Lösungsmöglichkeiten zu diskutieren. Hierdurch wird der Frage „*Welche Konzepte bestehen derzeit in der Erbringung des Sicherheitsnachweises für die eingesetzten ML-Arten in FAS?“* (siehe Unterkapitel 1.2) nachgegangen.

Diese Lösungskonzepte werden hinsichtlich offener Fragestellungen untersucht, um festzustellen, ob die Frage „*Welche Defizite besitzen diese Konzepte?*“ mit „Keine“ beantwortet wird oder ob hierdurch weitere Fragestellungen aufgeworfen werden, um die initiale Forschungsfrage „*Welche Herausforderungen bestehen im Nachweis der Sicherheit von ML in FAS?*“ umfassend zu beantworten.

3.1 Behandlung von ML in der ISO/ PAS 21448

Die ISO/ PAS 21448[111] steht im Einklang zu der Verwendung von ML, zumindest hinsichtlich offline gelernter Modelle, wie sie den Fokus der vorliegenden Arbeit darstellen (siehe Abschnitt 2.3.4). Vor allem im Bereich der Objekterkennung und –klassifikation werden Vorteile hinsichtlich der Reduktion von systematischen Fehlern durch die Nutzung von Lernverfahren erwartet. Im Anhang G der ISO/ PAS 21448, der als „informativ“ gekennzeichnet ist, wird beschrieben, dass die Datensammlung sowie der Entwicklungsprozess des gelernten Modells in Einklang zu nicht näher benannten Sicherheitsstandards zu entwickeln ist und hierdurch Gefahren, wie Verfälschungen der Realität in den Trainings-

[110] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[111] ISO: ISO/ PAS 21448 (2019).

daten, zu reduzieren sind. Außer der Forderung nach Sicherheitsstandards zu entwickeln, gibt die ISO/ PAS 21448 jedoch keine expliziten Regeln vor, sondern erteilt Ratschläge, durch die es möglich sei, ein sicheres Systemverhalten zu erzeugen.

Es wird vorgeschlagen, die Methoden, die zum Sicherheitsnachweis von ML genutzt werden, unter Einbeziehung von ISO 26262:2018 Teil 8 Abschnitt 11 zu entwickeln. Der benannte Absatz behandelt die Qualifizierung von Werkzeugen bzw. Prozessen, bei denen ein inkorrektes Verhalten zu Fehlern im mit diesem Werkzeug/ Prozess entwickelten System führt. Neben diesem Hinweis wird der generelle Entwicklungsprozess von offline gelernten Modellen vorgestellt und folgende explizite Ratschläge gegeben:

- Überprüfung manueller Label/ Annotationen
- Bei Nichterreichen bestimmter Leistungsfähigkeit im Trainingsprozess sind mehr Daten zum Training zu sammeln oder das Training zu verändern.
- Nutzung eines separat erfassten und gelabelten/ annotierten Testdatensatzes anstelle einer Untermenge des Trainingsdatensatzes
- Bei Nichterreichen bestimmter Leistungsfähigkeit im Testprozess sind mehr Daten zum Training zu sammeln oder das Training zu verändern.
- Nutzung von strukturierten Analysen, um mögliche Quellen von Verfälschungen innerhalb des Trainingsprozesses zu vermeiden. Zu analysieren sind beispielsweise die Abdeckungsrate und die Vielfalt von:
    - Datensammlung: Fahrzeugen und Fahrern, Strecken und Umgebungsbedingungen, strukturierten Testfällen
    - Annotationen/ Label
    - Überprüfung der Annotationen/ Label.

3.2 Behandlung von ML in der ISO 26262

Eine von Salay et al.[112] durchgeführte Analyse der ISO 26262 hinsichtlich Widersprüchen zwischen Anwendung der Norm und Verwendung von ML ergab fünf widersprüchliche Bereiche. Die durchgeführte Analyse bezieht sich jedoch auf die ISO 26262:2011[113], welche inzwischen durch eine aktuellere Version (ISO 26262:2018[110]) abgelöst wurde. Die angesprochenen Diskussionspunkte sind in dieser neuen Version ebenfalls enthalten, jedoch berührt die in der Zwischenzeit erschienene ISO/ PAS 21448 einige der angesprochenen Widersprüche, wodurch diese nicht mehr in der Ausführungsform von Salay et al. Be-

[112] Salay, R. et al.: An Analysis of ISO 26262 (2017).

[113] ISO: ISO 26262:2011. Road vehicles: Functional safety (2011).

stand haben. Diese Punkte werden an den entsprechenden Stellen diskutiert. Die Analyse fokussiert auf offline gelernte Modelle, die sich nicht während ihres Einsatzes verändern, wie sie auch in FAS ihren Einsatz finden (vgl. Abschnitt 2.3.4). Die von Salay et al.[112] identifizierten Widersprüche sind im Folgenden zusammengefasst:

- **Gefahrenidentifikation**: Durch die Nutzung von ML ist es möglich, dass neue Gefahrenarten auftreten, wie bspw. zu starkes Vertrauen in das System durch den Nutzer. Es ist möglich, dass dieses Vertrauen durch das Ausstrahlen von „Verständnis" durch das gelernte Modell bedingt wird. Die Gefahrenidentifikation der ISO 26262 fokussiert lediglich Gefahren, die durch ein Fehlverhalten des Systems hervorgerufen wird, nicht durch den Benutzer. Salay et al. rät daher zu einer Erweiterung der Vorschrift der Gefahrenidentifikation auf die Interaktion zwischen Nutzer und System.[114a] Dieser Aspekt wird inzwischen durch die ISO/ PAS 21448 aufgegriffen, die explizit Falschgebrauch des Systems durch den Nutzer adressiert, weshalb er keinen Widerspruch mehr darstellt.
- **Fehler und Auswirkungen**: Die ISO 26262 fordert die Nutzung von systematischen Fehleranalysen, um festzustellen wie mögliche Fehler zu sicherheitskritischen Auswirkungen führen. Durch den veränderten Entwicklungsprozess besitzen gelernte Modelle allerdings spezifische Fehlerarten, die durch die bisherigen Analysen nicht berücksichtigt werden, wodurch es möglich ist, sicherheitskritisches Verhalten hervorzurufen. Salay et al. fordern daher die Nutzung spezieller Werkzeuge, um ML-spezifische Fehler zu identifizieren.[114a] Solche ML-spezifischen Fehler werden inzwischen durch die ISO/ PAS 21448 adressiert, wobei lediglich wenig konkrete Ratschläge erteilt werden, um diese Fehler zu identifizieren.[115] Hierdurch besteht zwar kein Widerspruch mehr zwischen Norm und ML, jedoch ist kein konkreter Lösungsweg vorgegeben. Eine Fehlerart stellt dabei fehlende Generalisierbarkeit durch das induktive Vorgehen zur Wissensgenerierung der gelernten Modelle dar.
- **Nutzung von Trainingsdaten**: Die ISO 26262 setzt voraus, dass ein anforderungsbasiertes Programmieren angewendet wird und die Funktionsweise des Systems vollständig spezifiziert ist. Der vorgestellte Entwicklungsprozess von ML widerspricht dieser Voraussetzung, da an Stelle der funktionalen Anforderungen der Trainingsdatensatz zur Spezifikation des Modells tritt. Salay et al. schlagen vor, dass die Anforderung an eine vollständige Funktionsspezifikation weicher zu gestalten ist und zusätzliche Anforderungen an die Qualität von Trainingsdatensätzen zu stellen sind.[114b] Salay und Czarnecki[116] erarbeiten hierzu in einer auf der grundlegenden Analyse aufbauenden Veröffentlichung konkrete Vorschläge. Die ISO/

[114] Vgl. Salay, R. et al.: An Analysis of ISO 26262 (2017), a: S. 3; b: S. 3f.

[115] Siehe Unterkapitel 3.1.

[116] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018).

PAS 21448 greift auch diesen Aspekt durch den Anhang G auf, wodurch der Widerspruch gelöst wird. Allerdings wird lediglich dazu geraten eine Prozessfehlermöglichkeits- und –einflussanalyse zu nutzen, um mögliche Fehler in den gesammelten Daten zu identifizieren. Weitere Hinweise werden nicht gegeben, wodurch ebenfalls kein konkreter Lösungsweg zur Lösung der Problematik existiert, dass ein unvollständiger Datensatz zu einer unvollständigen Umsetzung des gelernten Modells führt.

- **Abstraktionsebene auf der ML eingesetzt wird**: Die ISO 26262 definiert, dass eine Softwarearchitektur aus Komponenten und ihren Interaktionen in einer hierarchischen Struktur zu bestehen hat. Diese Zerlegung in Komponenten besitzt eine Sicherheitsrelevanz, da sie die Verständlichkeit eines komplexen Systems erhöht und die Verwendung formaler Analyseverfahren für die Zusammensetzung ermöglicht.[114b] Maschinelles Lernen ist in der Lage, eine komplexe Aufgabe, wie beispielsweise die Ermittlung einer Trajektorie auf Basis von Kamerabildern, ganzheitlich zu lernen.[117] In diesem Fall existiert jedoch keine architekturunabhängige Entwicklung einzelner Komponenten mehr, da die Architektur mit der Entwicklung der einzelnen Teilaufgaben verbunden ist. Salay et al. empfehlen daher den Einsatz von maschinell gelernten Systemen lediglich auf Komponentenebene, um den Anforderungen der ISO 26262 zu genügen.[114b]
- **Softwarerichtlinien**: Durch die Verwendung von ML ist es lediglich möglich, ca. 40% der spezifischen Softwarerichtlinien, die in der ISO 26262 definiert sind, umzusetzen. Diese Softwarerichtlinien dienen in der Norm unter anderem dazu, die Qualität der Implementierung sicherzustellen. Salay et al. schlagen die Neudefinition der Richtlinien vor, so dass diese keine speziellen Methoden mehr enthalten, sondern die dahinterstehende Zielsetzung spezifiziert wird, damit die Richtlinien sowohl für konventionelle als auch für gelernte Komponenten anwendbar sind.[114b] Hierzu erarbeiteten Salay und Czarnecki[118] konkrete Vorschläge in einer weiterführenden Veröffentlichung (siehe Abschnitt 3.2.2.3).

Anhand dieser Analyse ist festzustellen, dass die identifizierten Widersprüche teilweise relativ einfach auszuräumen sind oder bereits durch die ISO/ PAS 21448 ausgeräumt wurden, wie durch die Beschränkung des Einsatzes von ML auf Komponentenebene. Teilweise, wie bei der Problematik der Nutzung bzw. der Erhebung der Trainingsdaten und der Erarbeitung von Softwarerichtlinien, besteht noch weiterer Forschungsbedarf, welcher z.T. bereits von Salay und Czarnecki[118] adressiert wurde. Allerdings fehlt bisher ein Praxisbeispiel, an welchem die Umsetzung der erarbeiteten Richtlinien erläutert wird.

[117] Vgl. Pomerleau, D. A.: ALVINN (1989).

[118] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018).

Es verbleibt auch unter der Voraussetzung, dass ein gelerntes Modell

- auf Komponentenebene
- als Ersatz einer bereits bestehenden Funktionalität,
- für welche eine vollständige Spezifikation vorliegt, eingesetzt wird
- und, dass für die einzuhaltenden Softwarerichtlinien Ersatzmethoden bekannt sind,

das Problem, dass es sich bei den in FAS eingesetzten ML um eine Black-Box handelt. Das Modell mit diesen Bedingungen wird im Folgenden als Fallbeispiel genutzt. Gelernte, leistungsfähige Modelle sind in der Regel für den Menschen schwer interpretierbar. Dies liegt unter anderem in der hohen Dimensionalität begründet, aus welcher die hohe Leistungsfähigkeit resultiert.[119] Durch die Beschränkung des menschlichen Vorstellungsvermögens auf wenige Dimensionen ist es nicht möglich, ein Verständnis für die Vorgänge und Zusammenhänge in einem hochdimensionalen, komplexen Algorithmus zu erhalten.[120] Hierdurch ist es nicht möglich, Fehler innerhalb der gelernten Zusammenhänge zwischen In- und Outputgrößen analytisch zu ermitteln. Der ebenfalls von Salay identifizierte Widerspruch, dass ML Fehlerarten besitzt, die bisherige Fehleranalysen nicht berücksichtigen, ist daher nicht auszuschließen, da auch durch die ISO/ PAS 21448 keine konkrete Vorgehensweise zur Identifikation aller Fehlerarten gegeben wird. Hieraus ergibt sich die Gefahr, dass das Modell im Rahmen der Trainings- und Testdaten korrekt und sicher funktioniert, jedoch im Betrieb funktionale Unzulänglichkeiten auftreten. Ein Beispiel für einen solchen Fehler stellt das Fokussieren eines gelernten Modelles auf mikroskopischen Strukturen anstelle der relevanten makroskopischen Zusammenhängen dar. Besteht die Aufgabe eines Modells darin, verschiedene Objekte und Tiere anhand von Bildern korrekt zu klassifizieren, ist es möglich, dass das Modell einen Leoparden am Muster seines Fells erkennt, jedoch nicht an seiner Körperform. Wenn dieses Fellmuster auf anderen Objekten, wie einem Sofa, erscheint, wird dieses fälschlicherweise als Leopard klassifiziert.[121] Wird mit dem Testdatensatz oder Testfällen kein Objekt mit einem Leopardenmuster überprüft, bleibt dieser Fehler unentdeckt.

Es bestehen verschiedene Lösungsmöglichkeiten, um der vorgestellten Problematik der ML-inhärenten Fehler bei Vorliegen einer Black-Box zu begegnen. Eine davon besteht darin, dass eine testfallbasierte Beweisführung erbracht wird, die zeigt, dass keine Fehler bzw. nur Fehler mit akzeptierten Auswirkungen im Modell existieren. Darüber hinaus besteht die Möglichkeit den ML-inhärenten Fehlern strukturiert auszuschließen bzw. einen anders gearteten Nachweis der Sicherheit der gelernten Modelle, trotz der ML-inhärenten Fehler, zu liefern. Beide Möglichkeiten werden im Folgenden vorgestellt und hinsichtlich ihrer Anwendbarkeit auf Fahrerassistenzsysteme diskutiert.

[119] Vgl. Nusser, S.: Dissertation, Robust Learning in Safety-Related Domains (2009), S. 2.

[120] Vgl. Olah, C.: Visualizing MNIST: An Exploration of Dimensionality Reduction (2014).

[121] Vgl. Khurshudov, A.: Suddenly, a leopard print sofa appears (2015).

3.2.1 Testfallbasierte Beweisführung

Im Fall einer Black-Box bedeutet eine testfallbasierte Beweisführung das vollständige Abtesten des im Betrieb möglichen Parameterraums, was zu einer sehr hohen Anzahl an notwendigen Testfällen führt.[122] Zudem sind die Ergebnisse des Tests lediglich für das überprüfte Modell in exakt dieser Variation gültig. Sobald sich Modellparameter ändern, sind die Testfälle erneut durchzuführen, da kein Wissen zu den Auswirkungen der Änderungen auf das Modellverhalten bzw. den möglichen Fehlverhalten besteht. Hierdurch ist das Black-Box-Testen unwirtschaftlich bzw. nicht praktikabel zur Erbringung eines Sicherheitsnachweises. Auf eine wissensbasierte Testfallerzeugung zur vollständigen Testraumabdeckung ist nicht zurückzugreifen, da im Fallbeispiel[123] zwar die Funktion und die, mit dieser Funktion verbundenen, möglichen Fehler bekannt sind, jedoch nicht die Fehler, die dem gelernten Modell inhärent sind, wie die beschriebene Fokussierung auf mikroskopische Strukturen. Hierdurch ist es nicht möglich, die Anzahl an notwendigen Testfällen zur Testraumabdeckung zu senken. Es resultiert die Notwendigkeit, nach der Suche von neuen Konzepten für die Erbringung des Nachweises der Sicherheit von ML in FAS, da die aktuelle verwendete Methodik keine Lösungsmöglichkeit für gelernte Algorithmen darstellt.

Drei bestehende Lösungsansätze für die Problemstellung der hohen Anzahl an notwendigen Testfällen wurden bereits zuvor veröffentlicht.[124] Alle haben zum Ziel, die Anzahl an notwendigen Testfällen bzw. den Aufwand zur Erbringung des Sicherheitsnachweises zu senken:

- Die Begrenzung des Arbeitsbereichs des Modells zur Verringerung der Testraumgröße
- Die Verbesserung der Interpretierbarkeit leistungsfähiger Modelle zur Ermöglichung eines (teilweisen) analytischen Sicherheitsnachweises
- Die Verbesserung der Leistungsfähigkeit interpretierbarer Modelle zur Ermöglichung eines (teilweisen) analytischen Sicherheitsnachweises

Die einzelnen Ansätze werden im Folgenden kurz vorgestellt. Weitere Details zu den Ansätzen sind Henzel et al.[124] zu entnehmen. Die folgenden Abschnitte (3.2.1.1 bis 3.2.1.3) wurden teilweise unverändert aus dieser Veröffentlichung entnommen.

3.2.1.1 Begrenzung des Wertebereichs

Durch Begrenzung des Wertebereichs von Eingangs-, Zustands- und Ausgangsgrößen auf Werte des Trainingsdatensatzes oder diese Werte umfassende Wertebereiche sowie eine

[122] Vgl. test IO: Black Box Testing (2019).

[123] Siehe Unterkapitel 3.2, Seite 30.

[124] Henzel, M. et al.: Herausforderungen in der Absicherung von FAS (2017).

dynamische Begrenzung der Änderungsrate dieser Größen ist es möglich, die Arbeitsgrenzen des Modells zu fixieren und hierdurch die Größe des Testraums zu verringern.[125a] [126]

3.2.1.2 Verbesserung der Interpretierbarkeit leistungsfähiger Modelle

Durch die Einschränkung des menschlichen Vorstellungsvermögens auf wenige Dimensionen ist es nicht möglich, ein Verständnis für die Vorgänge und Zusammenhänge in ein hochdimensionales, komplexes Modell zu erhalten.[127] Daher beruhen die Ansätze zur Verbesserung der Interpretierbarkeit eines solchen Modells auf dem Grundgedanken, dessen Dimensionalität zu reduzieren oder das im Modell verwendete Wissen explizit darzustellen:

- Herunterbrechen eines hochdimensionalen Algorithmus auf mehrere niedrigdimensionale Teilmodelle[128]
- Niedrigdimensionale Visualisierung der Vorgänge in einem komplexen Algorithmus zum Verständnis der Entscheidungen[129] [130] [131]
- Implementierung eines Zuverlässigkeitsmaßes[132]
- Extraktion von verwendeten Regeln aus dem gelernten Modell[133] [134]
- Explizite Begründung der Vorhersage[135]

Durch das Herunterbrechen eines hochdimensionalen Algorithmus in mehrere Teilmodelle, die menschlich interpretierbar sind, und das anschließende Verknüpfen dieser Teilmodelle bleibt die menschliche Interpretierbarkeit des Gesamtmodells erhalten. Die praktische Anwendbarkeit des Ansatzes im automobilen Bereich wurde bereits erwiesen, allerdings lediglich für einfache Klassifikationen. Hervorzuheben ist, dass es durch den Erhalt der vollständigen Interpretierbarkeit möglich ist, den Nachweis des fehlerfreien Verhaltens komplett analytisch zu erbringen.[128]

---

[125] Vgl. Otte, C.: Safe and Interpretable Machine Learning (2013), a: S. 113; b: S. 115f.

[126] Vgl. Henzel, M. et al.: Herausforderungen in der Absicherung von FAS (2017), S. 143.

[127] Vgl. Olah, C.: Visualizing MNIST: An Exploration of Dimensionality Reduction (2014).

[128] Vgl. Nusser, S.: Dissertation, Robust Learning in Safety-Related Domains (2009).

[129] Vgl. Olah, C. et al.: Feature Visualization (2017).

[130] Vgl. Yosinski, J. et al.: Understanding Neural Networks (2015).

[131] Vgl. Zhang, Q.-s.; Zhu, S.-c.: Visual interpretability for Deep Learning (2018).

[132] Vgl. Li, L. et al.: Knows what it knows: a framework for self-aware learning (2008).

[133] Vgl. Ramachandran, U. B.: Masterthesis, Issues in Verification and Validation of NN (2005).

[134] Vgl. Katz, G. et al.: Reluplex (2017).

[135] Vgl. Hendricks, L. A. et al.: Generating Visual Explanations (2016).

Die niedrigdimensionale Visualisierung der Vorgänge in einem gelernten Modell sowie die Nutzung eines Zuverlässigkeitsmaßes der Ausgangsgröße erhöhen das Verständnis über die Entscheidungen des Modells, sind aber nicht in der Lage eine solche Einsicht in das Modell zu gewähren, dass alle Fehler sicher identifiziert werden. Daher ist es nicht möglich, die Testfallanzahl mit diesen Ansätzen zu verringern.

Durch einen bestimmten Ansatz ist es möglich, aus einem NN mit einer bestimmten Art der Aktivierungsfunktion quantitative Beziehungen zwischen Eingangs- und Ausgangsgrößen nachzuvollziehen und hierdurch die vom Modell verwendeten Regeln zu extrahieren. Der Ansatz wurde im Bereich der Luftfahrt angewendet.[134] Darüber hinaus existiert ein weiterer Ansatz zur Extraktion von Regeln aus einem NN, der prototypisch im automobilen Bereich angewendet wurde.[133] Beide Ansätze wurden speziell für die jeweiligen Basis-Algorithmen entwickelt und sind nicht auf andere Algorithmen übertragbar.

Die Methode der expliziten Begründung der Vorhersage durch das gelernte Modell ist im Bereich der Bildklassifikation zu finden. Das Modell erläutert, warum es zu einer bestimmten Vorhersage kommt, beispielsweise im Rahmen der Klassifikation von Vögeln. „Es handelt sich um einen Renntaucher, da dieser Vogel einen langen weißen Hals, einen spitzen gelben Schnabel und ein rotes Auge hat“[136]. Es handelt sich hierbei um einen speziellen Algorithmus, der nicht auf andere Algorithmenarten übertragbar ist. Zudem erhöht der Ansatz zwar das Verständnis über die Entscheidungsfindung des gelernten Modells, jedoch ist es aber nicht möglich, hierdurch das Vorliegen von ML-inhärenten Fehlern auszuschließen.

3.2.1.3 Verbesserung der Leistungsfähigkeit interpretierbarer Modelle

ML-Algorithmen, die für ihre hohe Transparenzeigenschaft auf Basis ihrer Grundstruktur bekannt sind, sind beispielsweise Entscheidungsbäume oder Symbolic-Regression-Algorithmen, die zur Klassifikation oder Regression genutzt werden.[137] Allerdings ist deren Leistungsfähigkeit im Vergleich zu bspw. NN gering, weshalb sie in FAS nur selten, im Falle von Entscheidungsbäumen (siehe Unterkapitel 2.3), oder überhaupt nicht, wie im Falle von Symbolic-Regression-Algorithmen, verwendet werden. Zur Verbesserung deren beschränkter Leistungsfähigkeit wurden folgende Ansätze identifiziert:

- Verwendung von komplexeren Modellen in den Knoten von Entscheidungsbäumen[138]
- Verwendung von komplexeren Vorhersagemodellen für ganze Äste von Decision-Trees, die für die Erfüllung von Sicherheitszielen nicht relevant sind

[136] Übersetzt aus dem Englischen: Hendricks, L. A. et al.: Generating Visual Explanations (2016), S. 2.

[137] Vgl. Otte, C.: Safe and Interpretable Machine Learning (2013), S. 115f.

[138] Vgl. Loh, W.-Y.: Regression by Parts: Fitting Visually Interpretable Models with GUIDE (2008).

- Verbesserung der durch das einfache Design verbleibenden (Vorhersage-)Fehler durch komplexere Modelle[137]

Für den ersten und dritten Ansatz findet sich jeweils ein Anwendungsbeispiel, welches die erreichte Verbesserung der Leistungsfähigkeit gegenüber den verwendeten Basis-Algorithmen (einmal Entscheidungsbaum, einmal Symbolic-Regression) beziffert. Allerdings sind diese Beispiele außerhalb des automobilen Kontexts.[137] [138] Durch das Erhalten des interpretierbaren Basis-Algorithmus ist es möglich, den Nachweis des sicheren funktionalen Verhaltens der Modelle komplett analytisch zu führen. Der zweite Ansatz wurde bisher nicht erforscht und bedarf weiterer Untersuchungen hinsichtlich des Nutzens, da die Leistungsfähigkeit des Modells lediglich in Bereichen verbessert wird, welche nicht sicherheitskritisch sind.

3.2.1.4 Zusammenfassung

Einige Ansätze zur Reduktion des Testfallaufwandes für gelernte Algorithmen wurden bereits erarbeitet. Der erste Ansatz, die Begrenzung des Arbeitsbereichs, besitzt den Nachteil, dass abhängig von der Größe des Arbeitsbereichs dennoch eine hohe, abhängig von der Problemstellung teilweise unwirtschaftliche Anzahl an Testfällen durchzuführen ist, um die ML-inhärenten Fehler zu adressieren bzw. auszuschließen, da die Black-Box nicht „geöffnet“, sondern nur „verkleinert“ wird. Die Begrenzung des Arbeitsbereichs hat ebenfalls zur Folge, dass der Nutzen des Modells geschmälert wird. Zudem ist eine Begrenzung des Arbeitsbereichs nicht im Rahmen jeder Problemstellung durchführbar, wie beispielsweise in der Detektion von Objekten aus Bildern.

Die Ansätze, die darauf basieren, einen Kompromiss zwischen Interpretierbarkeit der gelernten Modelle bei gleichbleibender oder erhöhter Leistungsfähigkeit herzustellen, zeichnen sich dadurch aus, dass die Wahl des Ansatzes zur Reduktion der Testfälle bereits im Rahmen der Entwicklung des Modells zu treffen ist. Diese Ansätze benutzen spezielle Algorithmen, um diesen Kompromiss zu erreichen. Es ist nicht möglich, diese Ansätze nachträglich auf bereits gelernte Modelle anzuwenden. Hierdurch existiert keine allgemeingültige Lösung zur Reduktion der Testfallanzahl.

3.2.2 Alternativen zur testfallbasierten Beweisführung

Die Alternativen zum testfallbasierten Nachweis der Sicherheit des gelernten Modells trotz möglicher inhärenter Fehler lassen sich in zwei Kategorien unterteilen:

- Strukturierte Sicherheitsnachweise, die sich der Goal-Structuring-Notation bedienen
- Ausgangsgrößenbasierter Sicherheitsnachweis

Des Weiteren findet sich eine Sammlung von Ratschlägen, wie ein sicheres Systemverhalten zu erreichen ist. Im Folgenden werden die einzelnen Kategorien vorgestellt, sowie auf die Ratschläge eingegangen. Die Ansätze werden hinsichtlich ihrer Eignung zur vollständigen Erbringung des Sicherheitsnachweises in einer Zusammenfassung in Abschnitt 3.2.2.4 bewertet.

3.2.2.1 Strukturierte Sicherheitsnachweise

Im Rahmen einer Literaturrecherche wurden drei Ansätze strukturierter Sicherheitsnachweise für die Verwendung maschineller Lernverfahren im Rahmen von sicherheitskritischen System identifiziert. Sie bedienen sich alle der Goal-Structuring-Notation (GSN), welche zur graphischen Darstellung einer Argumentationskette genutzt wird. Sie stellt explizit die einzelnen Elemente eines Sicherheitsnachweises, wie Beweise, Anforderungen, Voraussetzungen, Lösungen etc. und deren Beziehung zueinander dar.[139] Kurd[140] [141] beschäftigt sich explizit mit dem Sicherheitsnachweis für ein Neuronales Netz, welcher domänenunabhängig formuliert ist. Die anderen beiden Ansätze von Rudolph et al.[142] und Burton et al.[143] sind prinzipiell nicht algorithmenspezifisch, beschränken ihre Betrachtung jedoch auf offline gelernte, deterministische Lernansätze, die ebenfalls Neuronale Netze umfassen. Beide Ansätze nehmen an, dass das gelernte Modell für eine Funktion im Rahmen eines automatisierten Fahrens zuständig ist.

Kurdscher Sicherheitsnachweis

Kurd[140] analysiert ein NN zur Erstellung des Nachweises und definiert ein Minimum an notwendigen Eigenschaften für ein sicheres Verhalten des Algorithmus. Für die Ableitung der Eigenschaften findet sich kein Nachweis eines Top-Down-Vorgehens oder eine Begründung, dass diese notwendigen Eigenschaften auch hinreichend für einen Sicherheitsnachweis sind.[144] Es wird allerdings gezeigt, dass die Verletzung der geforderten Eigenschaften aus einer Vielzahl an möglichen Fehlern resultieren, wodurch das Vorliegen dieser Eigenschaften diese Fehler ausschließt.[145a] Diese Eigenschaften sind:

- G2: Die Input-Output-Funktionen für das NN wurden sicher zugeordnet.
- G3: Das beobachtbare Verhalten des NN muss vorhersehbar und wiederholbar sein.

[139] Vgl. The Assurance Case Working Group: Goal Structuring Notation (2018), S. 11ff.

[140] Kurd, Z.: Dissertation, Neural Networks in Safety-critical Applications (2002).

[141] Kurd, Z. et al.: Developing neural networks for safety critical systems (2006).

[142] Rudolph, A. et al.: A consistent safety case argumentation for artificial intelligence (2018).

[143] Burton, S. et al.: Case for Safety of Machine Learning (2017).

[144] Vgl. Kurd, Z.; Kelly, T.: Establishing Safety Criteria for NN (2003).

[145] Vgl. Kurd, Z. et al.: Developing neural networks for safety critical systems (2006), a: S. 14; b: S. 11f.

- G4: Das NN toleriert Fehler in seinen Eingängen.
- G5: Das NN erzeugt keine gefährlichen Ausgänge.

„Sicher“ im Rahmen der Eigenschaft G2 bezieht sich auf die Minderung oder Beherrschung potentieller Gefahren, die in Verbindung mit den Eigenschaften der Input-Output-Zuordnungen möglich sind.[145b] Durch die Eigenschaft G2 wird erreicht, dass die vom NN ausgeführte Funktion auch die gewünschte Funktion darstellt. Um diese Eigenschaft nachzuweisen, werden analytische Methoden wie Dekompositionsansätze vorgeschlagen. Diese versuchen das Verhalten des NN durch Analyse der Struktur des Netzes zu erhalten, um die Black-Box in eine White-Box zu verwandeln. Kurd gibt hierfür jedoch kein Anwendungsbeispiel oder detaillierte Hinweise zur Durchführung dieser Methoden, sondern stellt heraus, dass noch offene Forschungsfragen hinsichtlich der Erläuterung der Outputs von NN sowie dem Generalisierungsverhalten existieren.[146] Für die anderen geforderten Eigenschaften gibt Kurd ebenfalls Ratschläge oder Ansätze, wie deren Existenz zu beweisen ist, führt diese jedoch nicht konkret durch und lässt hierdurch Fragestellungen offen.

Rudolphscher Sicherheitsnachweis

Rudolph et al.[142] setzen voraus, dass eine Gefahrenanalyse des Systems entsprechend den Vorgaben der ISO 26262[147] vorliegt und beziehen die Argumentation in ihrem Sicherheitsnachweis auf die Ergebnisse dieser Analyse. Eine Beschreibung, wie die Struktur des Nachweises abgeleitet wurde, ist in der Veröffentlichung nicht gegeben. In Abbildung 3-1 ist die Hauptstruktur des Nachweises dargestellt, an welche sich an den Stellen G4 und G5 weitere Äste angliedern. In der genutzten GSN wird mit „C“ der Kontext, in dem die Forderungen bzw. Ziele „G“ aufgestellt sind, markiert. „S“ bezeichnet eine Strategie, die zur Erfüllung der Forderungen bzw. Ziele genutzt wird. Mit „A“ sind Annahmen und mit „J“ sind Rechtfertigungen gekennzeichnet.[139] „Sn“ stellt Lösungen zur Erfüllung des Ziels bzw. der Forderung dar.

[146] Vgl. Kurd, Z.; Kelly, T.: Establishing Safety Criteria for NN (2003), S. 4f.

[147] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

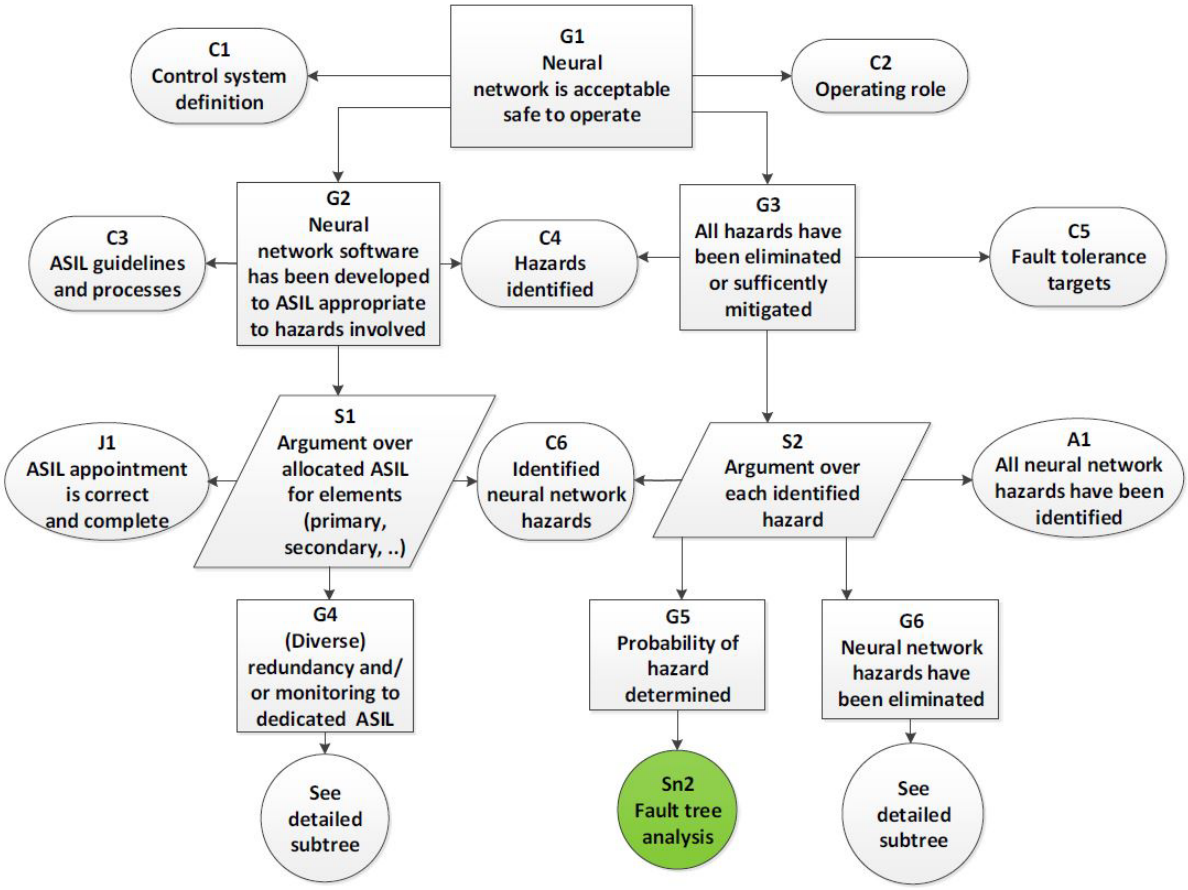


Abbildung 3-1: Sicherheitsnachweis eines Neuronalen Netzes (Auszug)[148]

Der Grundgedanke des Ansatzes besteht darin, dass systematisch alle möglichen Gefahren entweder akzeptabel gemildert oder ganz vermieden werden. Hierzu werden angegliedert an G4 Methoden zur Überwachung oder zur Erlangung von Redundanz bzw. Fehlertoleranz vorgeschlagen. An G6 schließt sich die Begründung an, wie durch eine robuste Netzwerktopologie (G7), der Überwachung des zeitlichen Verhaltens des Modells im Betrieb (G9), der Beschränkung von Ausgangsgrößen (G10) und dem Einsatz formalen Methoden (G11), der Nachweis, dass alle Gefahren des Netzes gebannt wurden (G6), geführt wird. Es handelt sich hierbei um eine Auflistung verschiedener Werkzeuge (siehe Abbildung 3-2 für den Unterast G8 inkl. G10 und G11), wobei der Nachweis, dass diese Werkzeuge zur Erfüllung der einzelnen Forderungen bzw. Ziele führen, noch zu erbringen ist.

[148] Rudolph, A. et al.: A consistent safety case argumentation for artificial intelligence (2018), S. 6.

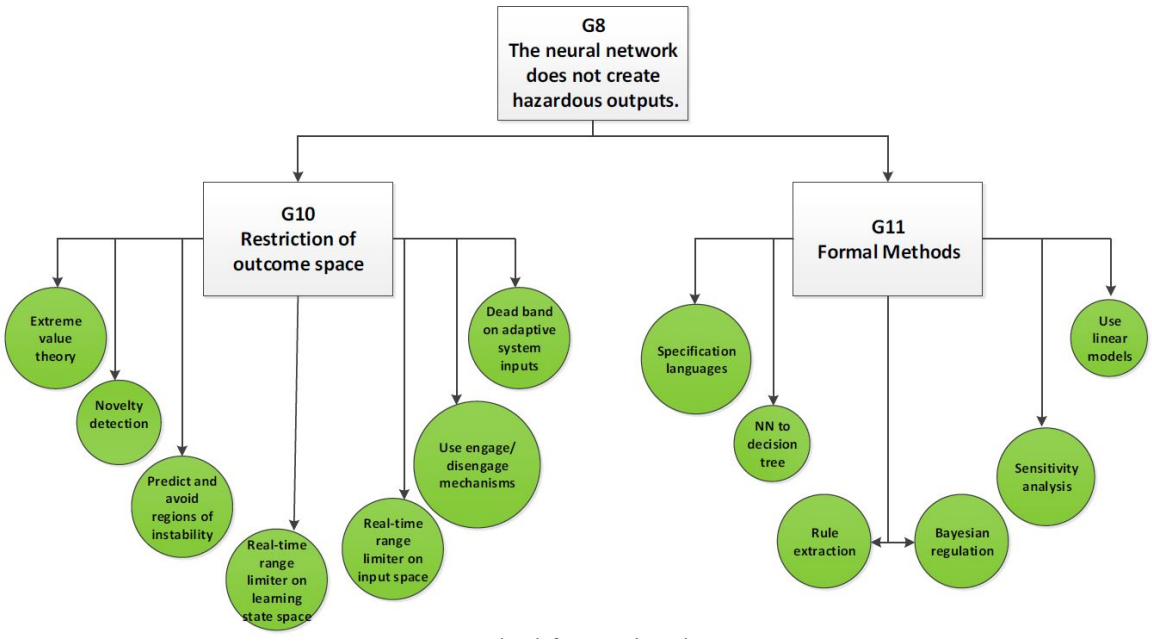


Abbildung 3-2: Sicherheitsnachweis eines Neuronalen Netzes, Unterast von G6[148]

Darüber hinaus besitzt der Sicherheitsnachweis die Annahme, dass alle Gefahren, die vom Algorithmus ausgehen, identifiziert wurden (siehe A1 in Abbildung 3-1). Dies schließt ebenfalls die ML-inhärenten Gefahren ein sowie Gefahren, die durch das Ausstrahlen eines „intelligenten“ Systemverhaltens hervorgerufen werden. Dieser Annahme entspricht allerdings nicht dem aktuellen Forschungsstand, da bisher keine Auflistung aller ML-inhärenten Fehler identifiziert wurde. Generell ist der Sicherheitsnachweis von Rudolph et al. auf andere offline gelernte, deterministische Algorithmen anwendbar, wobei die vorgeschlagenen Werkzeuge zur Beweiserfüllung hinsichtlich ihrer Übertragbarkeit zu überprüfen sind.

Burtonscher Sicherheitsnachweis

Der Kern des Sicherheitsnachweises von Burton et al.[149] besteht in der Strategie, dass die verschiedenen Gründe für funktionale Unzulänglichkeiten des gelernten Modells, also die ML-inhärenten Fehler, systematisch ausgeschlossen werden. Ob, mit welcher Systematik und mit welcher Vollständigkeit die Gründe identifiziert wurden, ist nicht publiziert. Diese Gründe werden im Rahmen des Sicherheitsnachweises als Ziele bzw. Forderungen definiert:

[149] Burton, S. et al.: Case for Safety of Machine Learning (2017).

- G2: Die Einsatzbedingungen sind hinreichend definiert und in den Trainingsdaten wiederzufinden.
- G3: Die Funktion ist robust gegenüber einer Verteilungsverschiebung der Eingangs- und Ausgangsgrößen zwischen Training- und Betriebsdaten.
- G4: Die Funktion verhält sich gleich in kritischen Situationen der gleichen Art.
- G5: Die Funktion ist robust gegenüber Unterschieden zwischen der Datenerhebungs- und Betriebsplattform (bspw. Änderungen in den Vorverarbeitungsschritten der Eingangsdaten).
- G6: Die Funktion ist robust gegenüber Änderungen der Systembedingungen (bspw. Änderungen in der Erfassungshardware).

Burton et al.[149] diskutiert anschließend Möglichkeiten, um diese Forderungen zu erfüllen und identifiziert fünf mögliche Bereiche:

- **Abdeckungsrate der Trainingsdaten**: Es sind Kriterien zu generieren, die die ausreichende Menge an Eingangsdaten für eine bestimmte Funktionalität feststellen.
- **Interpretierbarkeit der gelernten Funktion**: Es sind Methoden anzuwenden, die die Interpretierbarkeit der gelernten Modelle erhöhen, um Fehler zu identifizieren.
- **Berechnung der Unsicherheit**: Die Quantifizierung der Unsicherheit kann Informationen liefern, die beispielsweise in Plausibilitäts- und Sensorfusionsalgorithmen verwendet werden, wodurch die Robustheit und Zuverlässigkeit der nachfolgenden Aufgaben wie Trajektorienplanung insgesamt verbessert wird.
- **Black-Box Testmethoden**: Da konventionelle White-Box-Testmethoden zur Überprüfung der Funktion nur sehr begrenzt auf ML anwendbar sind, ist ein Fokus auf die Entwicklung und Anwendung von Black-Box-Techniken zu legen.
- **Echtzeitmaße**: Eine weitere Beweisquelle, um Auswirkungen von inkorrekter Funktionalität zu minimieren, ist die Echtzeitüberwachung von Kriterien, wie eine Plausibilitätsüberprüfung der Ausgänge des gelernten Modells mit konventionellen Modellen oder die Echtzeitüberprüfung von Anforderungen, die im Betrieb einzuhalten sind.

Burton et al. trifft selbst die Aussage, dass sein Sicherheitsansatz keinen umfassenden Sicherheitsnachweis darstellt, sondern den Rahmen für weitere Entwicklungen gibt.[150]

3.2.2.2 Ausgangsgrößenbasierter Sicherheitsnachweis

Die zentrale Idee, die hinter dem ausgangsgrößenbasierten Sicherheitsnachweis von Faria[151] steht, ist, dass für jede mögliche Variation der Ausgangsgrößen zu identifizieren und

[150] Vgl. Burton, S.; Bürkle, L.: Making the Case for Safety of Machine Learning (2017), S. 61.

zu beweisen ist, dass deren Auswirkung die Sicherheit des Gesamtsystems nicht beeinflusst. Es wird sich daher dem Konzept des Determinismus bedient, um sicheres Verhalten des gelernten Modells zu erhalten. Hierzu wird folgendes Vorgehen vorgeschlagen:

1. Jede Variation des Ausgangs ist zu identifizieren (algorithmenunabhängig, da die Variationen aufgrund von Fehlern in den gelernten Modellen auftreten).
2. Die Auswirkungen von 1. sind zu untersuchen (systemabhängig).
3. Es sind Kontrollmechanismen zu definieren, die das Einhalten von Sicherheitsgrenzen trotz 2. sicherstellen (systemabhängig).

Faria[151] vertieft in der zugehörigen Veröffentlichung den ersten Schritt und identifiziert Ursachen für Fehlverhalten. Dabei wird kein Anspruch auf Vollständigkeit der identifizierten Ursachen erhoben, sondern herausgestellt, dass die Auflistung zu erweitern ist. Die von ihm vorgestellten Ursachen werden in fünf Kategorien gegliedert:

- **Erfahrung**:
  - Die Repräsentativität der Trainingsdaten ist zu gering.
  - Die Label sind nicht korrekt.
  - Es sind nicht genügend Trainingsdaten zur Extraktion der relevanten Zusammenhänge vorhanden.
- **Aufgabe**: Die Aufgabe ist nicht generell genug gelernt.
- **Algorithmus:**
  - Bei nicht-konvexen Kostenfunktionen wird lediglich ein lokales Minimum erreicht, nicht das globale.
  - Eine falsche Wahl der Modellparameter wie beispielsweise der Clusteranzahl.
- **Implementierung**: Die genutzte Programmiersprache enthält vereinfachte Berechnungen (bspw. bei der Berechnung von Matrixinversen).
- **Hardware**: Die genutzte Hardware zur Berechnung verursacht Fehler.

Das vorgeschlagene Vorgehen ist prinzipiell auf alle für FAS relevanten ML-Arten übertragbar.

3.2.2.3 Ratschläge zur Erlangung sicherer Systeme

Neben strukturierten Ansätzen zum Nachweis der Sicherheit bei der Verwendung gelernter Modelle finden sich generelle Ratschläge, wie diese Sicherheit zu erreichen ist. Var-

[151] Faria, J. M.: Non-determinism and Failure Modes in Machine Learning (2017).

shney[152] stellt in seiner Veröffentlichung vier verschiedene, domänenübergreifende Ansätze vor, mit denen es möglich ist, die Sicherheit zu erhöhen. Diese Ansätze werden im Folgenden vorgestellt und durch Aspekte einer Sammlung von Ratschlägen von Faria[153] und Lisboa[154] ergänzt:

- **Systementwickelung mit systeminhärenter Sicherheit**: Durch Ausschließen von allen potentiellen Gefahren ist es möglich, ein sichereres Systemverhalten zu erhalten. Eine Methode besteht darin, interpretierbare Modelle zu generieren.[155] Im Rahmen einer anderen von Varshney[152] vorgeschlagenen Methode werden Merkmale, die nicht ursächlich für den gewünschten Zusammenhang sind, ausgeschlossen. Als Beispiel für einen Algorithmus, in welchem beide Methoden vereint werden, sind Causal-Falling-Rule-Lists[156] genannt. Diese Methoden verschlechtern die Leistung eines gelernten Modells, verringern jedoch das Risiko von wissensbasierter Unsicherheit und erhöhen damit die Sicherheit. Darüber hinaus wird durch die Verwendung von Redundanzen die systeminhärente Sicherheit erhöht. Ein Beispiel hierfür ist das Trainieren mehrerer Modelle (sog. Ensembles), wobei resultierend die Vorhersage genutzt wird, die die meisten Modelle mit hoher Zuverlässigkeit ausgeben.[157] [158] Unter diesen Bereich fallen ebenfalls Lernalgorithmen, die explizites Wissen, welches durch den Entwickler vorgegeben wird, nutzen und sich hierzu lediglich zusätzliches Wissen aneignen.[158]
- **Einsatz von Sicherheitsreserven**: In Anlehnung an den klassischen Maschinenbau besteht eine Strategie zur Erhöhung der Sicherheit in der Entwicklung von gelernten Modellen mit Sicherheitsreserven. Die Sicherheitsreserve ist bei ML gekennzeichnet durch den Abstand zwischen dem tatsächlichen Risiko und dem angenommenen Risiko für die Auslegung. Diese Definition umfasst sowohl die Unsicherheiten die durch die Verteilung der Datenbasis als auch die durch die richtige Zuordnung der vorhandenen Daten hervorgerufen werden.
- **Einsatz von Rückfallebenen**: Wenn die Vorhersage des gelernten Modells eine geringe Zuverlässigkeit besitzt, sind Rückfallebenen wie konventionelle Modelle oder die Rückgabe an den Menschen vorzusehen. Allerdings besitzt die Wahl der korrekten Zuverlässigkeitsmetrik eine hohe Bedeutung, da bestimmte Metriken eine hohe Zuverlässigkeit bei falschen Vorhersagen liefern, wenn diese Vorhersagen aus

[152] Varshney, K. R.: Engineering safety in machine learning (2016).

[153] Faria, J.: Machine Learning Safety (2018).

[154] Lisboa, P. J.: Industrial use of safety-related artificial neural networks (2001).

[155] Vertiefend hierzu siehe Abschnitt 3.2.1.2 und 3.2.1.3.

[156] Wang, F.; Rudin, C.: Causal Falling Rule Lists (2017).

[157] Vgl. Lisboa, P. J.: Industrial use of safety-related artificial neural networks (2001), S: 23.

[158] Vgl. Faria, J.: Machine Learning Safety (2018), S. 16.

Eingangsdaten resultieren, deren Verteilungsdichte in den Trainingsdaten niedrig ist. Eine Möglichkeit diesem Problem zu begegnen, besteht in der sog. novelty detection, bei welcher bewertet wird, wie wahrscheinlich die aktuellen Eingangsgrößen im Betrieb zu der Verteilung der Trainingsdaten gehören.[157]

- **Einsatz von Prozess-Sicherheitswächtern**: Durch den Einsatz von Audits, Trainings und Warnungen ist es ebenfalls möglich, die Sicherheit im Systemdesign zu verbessern und beispielsweise fehlerhafte Programmierung zu vermeiden.

Neben dieser allgemeingültigen, domänenübergreifenden Sammlung stellen auch die aus der Analyse der Widersprüche zwischen der ISO 26262 und ML hervorgehenden Vorschläge für angepasste Richtlinien von Salay und Czarnecki[159] bzw. Salay et al.[160] Ratschläge zur Erlangung eines sicheren Systemverhaltens dar. Im Folgenden werden die Ratschläge von Salay und Czarnecki vorgestellt, die nicht bereits in der Veröffentlichung von Salay et al. enthalten sind, da diese bereits in Unterkapitel 3.2 erläutert werden.

- **Beginn Systemlebenszyklus**: Wenn sich eine sicherheitsrelevante Funktionalität neben einer Umsetzung mit ML auch (teilweise) mittels konventioneller Programmierung umsetzen lässt, ist die (teilweise) konventionelle Programmierung aufgrund der bestehenden Erfahrungen und Interpretierbarkeit gelernten Modellen vorzuziehen.
- **Anforderungsdefinition**: Durch die Problematik bedingt, dass ML dort eingesetzt wird, wo keine vollständige Funktionsspezifikation möglich ist, sind Teil-Verhaltensanforderungen sowie Datensatzanforderungen für die gelernten Funktionen aufzustellen. Unter Teil-Verhaltensanforderungen verstehen Salay und Czarnecki Bedingungen, die alle Ein- und Ausgangsgrößenpaare der Funktion zu erfüllen haben. Es werden verschiedene Arten der Teil-Verhaltensanforderungen vorgestellt, u.a. Invarianz und Äquivarianz. Die verschiedenen Arten werden in Unterkapitel 5.4 genauer erläutert. Die Erfüllung dieser Anforderungen gilt es im entwickelten Modell nachzuweisen. Es besteht ebenfalls die Empfehlung, dass Teil-Verhaltensanforderungen bereits als explizites „Vorwissen“ im Trainingsprozess vorgegeben werden.
- **Datensatzerweiterung**: Um die Anforderungen einzuhalten, wird vorgeschlagen synthetische Daten zur Integration in den Trainingsdatensatz zu generieren, die die Verhaltensanforderungen implizit enthalten. Zusätzlich dient die Veränderung bestehender Datenpunkte, bspw. hinsichtlich des Wetters, dazu, die Abdeckrate des Datensatzes zu erhöhen. Hierdurch ist es möglich, die Anzahl der sog. „bekannten Unbekannten“ zu vermindern. Diese stellen Datenpunkte dar, die durch Variation aus den bestehenden Daten aufgrund einer hohen Ähnlichkeit generiert werden

[159] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018).

[160] Salay, R. et al.: An Analysis of ISO 26262 (2017).

können. Die zugehörigen Label können aufgrund der Ähnlichkeit abgeleitet werden. Um die sog. „unbekannten Unbekannten“, was Datenpunkte sind, die keine Ähnlichkeit zu bestehenden Datenpunkten aufweisen und daher kritische Fälle für die Sicherheit des Systems darstellen, zu adressieren, wird die Identifikation gering abgedeckter Bereiche des Datensatzes vorgeschlagen. Hierauf basierend ist beispielsweise eine gezielte Sammlung neuer Daten für diese Bereiche möglich.

- **Architekturmaßnahmen**: Wenn die Sicherheit der Komponente, die ein gelerntes Modell enthält, nicht sichergestellt werden kann, ist es möglich, Fehlertoleranzmaßnahmen auf Architekturebene zu implementieren. Dabei sind die in der ISO 26262 vorgeschlagenen Methoden anwendbar. Darüber hinaus werden noch weitere Maßnahmen zur Fehlertoleranz vorgeschlagen.
- **Modellauswahl**: Wie bereits in anderen Publikationen, wird auch hier die Wahl von möglichst interpretierbaren Modellen gefordert, um maximale Interpretierbarkeit bei notwendiger Leistungsfähigkeit zu erhalten.
- **Auswahl der Features**: Die genutzten Eingangsgrößen des Modells haben eine kausale Beziehung zur Ausgangsgröße zu besitzen, nicht lediglich eine Korrelation.
- **Vermeidung von Fehlern im Trainingsprozess**: Die Bedingungen zwischen Trainingsdaten und späteren Betriebseingangsdaten sind möglichst gleich zu halten. Änderungen der Hardware zur Aufnahme der Daten sind bspw. zu vermeiden. Veränderungen in den Umgebungsbedingungen, wie z.B., dass sich eine Landschaft über Jahre hinweg verändert, sind in das Training einzubeziehen. Des Weiteren ist Overfitting[161] sowie die zu geringe Bestrafung von unsicherem Modellverhalten im Rahmen des Trainingsprozesses zu vermeiden.
- **Validierung und Test**: Es ist die Qualität der Generalisierung des gelernten Modells zu bestimmen. Zusätzlich ist jede einzelne inkorrekte Ausgabe des gelernten Modells im Rahmen des Testdatensatzes zu analysieren. Im Rahmen von Hardwaretests ist eine Fehlerrate ungleich Null aufgrund der zufällig möglichen Hardwarefehler akzeptabel. Im Rahmen von Softwaretests ist es jedoch möglich, alle Fehler im entwickelten Modell zu eliminieren, weshalb das auch zu erzielen ist. Das schließt dennoch nicht aus, dass Fehler des Softwaremoduls bei unbekannten Ereignissen auftreten. Jedoch wird diese Möglichkeit reduziert, wenn die Software auf dem Testdatensatz ein fehlerfreies Ergebnis liefert.

3.2.2.4 Zusammenfassung

Die vorgestellten Möglichkeiten zum Nachweis eines sicheren Modellverhaltens unter Berücksichtigung der ML-inhärenten Fehlerquellen liefern prinzipielle Vorgehensweisen und

[161] Detaillierte Erläuterung von Overfitting in Abschnitt 4.1.2, E22: Overfitting / Underfitting.

Vorschläge, wie ein sicheres Verhalten nachzuweisen bzw. zu erzeugen ist. Jedoch liefert keine der identifizierten Möglichkeiten einen umfassenden Nachweis sicheren Systemverhaltens. Eine wie durch die ISO/ PAS 21448[162] geforderte Qualifizierung der einzelnen Werkzeuge nach ISO 26262[163] Teil 8 Abschnitt 11 liegt ebenfalls nicht vor.[164] Allerdings bezieht diese sich auch auf tatsächliche Werkzeuge, die in Software implementiert sind und keine allgemeinen Vorgehensweisen zum Nachweis der Sicherheit, weshalb lediglich die Werkzeuge, die Rudolph et al.[165] auf unterster Ebene vorschlägt, zu qualifizieren sind.

Es existieren in den mittels GSN-strukturierten Sicherheitsnachweisen (Abschnitt 3.2.2.1) sowie dem ausgangsgrößenbasierten Sicherheitsnachweis (Abschnitt 3.2.2.2) offene Fragestellungen hinsichtlich der praktischen und durchgängigen Anwendbarkeit der Ansätze und der vollständigen Identifikation der ML-inhärenten Fehlerquellen, die beispielsweise von Rudolph et al.[166] angenommen wird. Kurd[167], Burton[168] und Faria[169] treffen selbst die Aussage, dass die vorgestellten Ansätze weiterer Forschung bedürfen. Die Ansätze von Rudolph et al. und Faria besitzen das größte Potential zur Anwendung für einen Sicherheitsnachweis, da ihre Struktur bei Nachweis der vollständigen Identifikation aller ML-inhärenten Fehlerquellen unverändert bleibt und lediglich die Anwendbarkeit der Ansätze zu zeigen ist.

Die vorgestellten Ratschläge (Abschnitt 3.2.2.3) verhelfen zu sicherem Systemverhalten, schließen ML-inhärente Fehler jedoch nicht explizit aus. Sie verhindern entweder die Auswirkungen von Fehlverhalten, beispielsweise durch den Einsatz von Rückfallebenen oder vermeiden das Auftreten der Fehler beispielsweise durch den Ausschluss von für die Problemstellung irrelevanten Merkmalen oder der Erhöhung der Qualität der Trainingsdaten.

3.2.3 Zusammenfassung der Lösungsmöglichkeiten

Aufgrund der nicht in konventionellen Algorithmen vorkommenden neuen ML-inhärenten Fehlerquellen in Kombination mit der fehlenden Interpretierbarkeit der gelernten Modelle, sind konventionelle Methoden zur Führung eines analytischen Sicherheitsnachweises nicht anwendbar. Eine rein testfallbasierte Beweisführung durch Black-Box-Testen ist aufgrund des großen Testraums ohne Kenntnis aller potentiellen Fehlerbereiche nicht wirtschaftlich.

---

[162] ISO: ISO/ PAS 21448 (2019).

[163] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[164] Siehe Unterkapitel 3.1.

[165] Rudolph, A. et al.: A consistent safety case argumentation for artificial intelligence (2018).

[166] Rudolph, A. et al.: A consistent safety case argumentation for artificial intelligence (2018).

[167] Kurd, Z. et al.: Developing neural networks for safety critical systems (2006).

[168] Burton, S. et al.: Case for Safety of Machine Learning (2017).

[169] Faria, J. M.: Non-determinism and Failure Modes in Machine Learning (2017).

Daher ist zur Beantwortung der Frage „*Welche Konzepte bestehen derzeit in der Erbringung des Sicherheitsnachweises für die eingesetzten ML-Arten in FAS?“* die Identifikation neuer Konzepte notwendig. Einerseits besteht die Möglichkeit, die notwendige Testfallanzahl zu reduzieren. Hierzu stehen die Begrenzung des Arbeitsbereichs sowie das Finden eines Kompromisses zwischen Leistungsfähigkeit und Interpretierbarkeit gelernter Modelle zur Verfügung. Die erste Möglichkeit „verkleinert“ dabei lediglich die Black-Box, wodurch je nach verbleibendem Modellraum eine noch immer hohe Anzahl an Testfällen resultiert. Die Entscheidung zur Nutzung der zweiten Möglichkeit ist bereits vor der Entwicklung der gelernten Modelle zu fällen, da diese Kompromisse algorithmenabhängig sind und die Wahl bestimmter Algorithmen erfordern. Durch die zweite Möglichkeit wird ein teilweiser analytischer Nachweis der Sicherheit ermöglicht. Andererseits besteht neben der testfallbasierten Beweisführung die Möglichkeit durch das strukturierte Ausschließen der ML-inhärenten Fehlerquellen einen Sicherheitsnachweis zu erbringen. Hierzu wurden vier verschiedene Sicherheitsnachweise vorgestellt, die jedoch in ihrer aktuellen Form alle keinen vollständigen Ausschluss der potentiellen Fehler gewährleisten. Auch Ratschläge zur Erlangung eines sicheren Systemverhaltens führen keinen vollständigen Nachweis.

Die Forschungsfrage „*Welche Konzepte bestehen derzeit in der Erbringung des Sicherheitsnachweises für die eingesetzten ML-Arten in FAS?*“ (vgl. Unterkapitel 1.2) lässt sich daher wie folgt beantworten:

Es existieren Konzepte zur Testfallreduktion, die entweder den Arbeitsbereich des gelernten Modells begrenzen oder auf menschlich interpretierbare Algorithmen zurückgreifen. Eine Alternative zur Testfallreduktion bieten Konzepte zur Erbringung eines strukturierten Sicherheitsnachweises.

Die Frage „*Welche Defizite besitzen diese Konzepte?*“ ist, wie bereits beschrieben, mit:

Der Ansatz der Arbeitsbereichbegrenzung resultiert in einem verminderten Nutzen des Modells bei gleichzeitiger Notwendigkeit innerhalb dieses Arbeitsbereichs aufwändige Black-Box-Testmethoden anzuwenden.

Durch den Einsatz menschlich interpretierbarer Modelle ist die Leistungsfähigkeit der resultierenden Modelle eingeschränkt. Zusätzlich ist dieser Ansatz nur anwendbar, wenn die Entscheidung für diese Art des Sicherheitsnachweises vor der Entwicklung getroffen wurde.

Die Konzepte für einen strukturierten Sicherheitsnachweis sind zwar prinzipiell für alle eingesetzten ML-Arten in FAS anwendbar, allerdings sind sie derzeit nicht in der Lage, alle ML-inhärenten Fehlerquellen auszuschließen

zu beantworten.

3.3 Konkretisierung weiterer Forschungsfragen

Durch die Antwort der Forschungsfrage „*Welche Defizite besitzen diese Konzepte?*“ aus Unterkapitel 1.2 ist zur weiteren Beantwortung der initiale Forschungsfrage „*Welche Herausforderungen bestehen im Nachweis der Sicherheit von ML in FAS?*“ die Analyse der ML-inhärenten Fehlerquellen bzw. Fehlerarten notwendig, um hierauf basierend weitere konkrete Forschungsfragen abzuleiten. Zur Identifikation der ML-inhärenten Fehlerquellen, die nicht in konventionellen Algorithmen auftreten, werden die Unterschiede zwischen beiden Algorithmenarten untersucht. Es werden entsprechend des Lebenszyklus von Algorithmen in FAS Unterschiede in den Bereichen Entwicklung der Modelle, Implementierung im Fahrzeug und Interaktion mit dem Nutzer untersucht.

Konventionelle Algorithmen werden entsprechend der ISO 26262[170] anforderungsbasiert entwickelt. Jede Funktionalität wird explizit vorgegeben und einprogrammiert. Dabei werden auch die Regeln, die zur Umsetzung der Funktionalität notwendig sind, als Anforderungen definiert und deren korrektes Verhalten im Anschluss überprüft. Diese Regeln werden im Vorfeld von Entwicklern basierend auf Messergebnissen, bestehenden Untersuchungen, analytischen Überlegungen etc. definiert und stellen für den gesamten Betriebsbereich gültige Zusammenhänge dar. Eine beispielhafte Regel für die Umsetzung einer Fahrstilklassifizierung basierend auf Fahrdynamikgrößen ist, dass sportlichere Fahrer andere Bereiche von Quer- und Längsbeschleunigungen erreichen als vorsichtigere.[171] Wie bereits im Rahmen des Entwicklungsprozesses maschinellen Lernens vorgestellt (siehe Abschnitt 2.2.1), wird das gelernte Modell im Gegensatz hierzu automatisiert durch das Induktionsprinzip erstellt.[172] Die zur Aufgabenerfüllung erforderlichen Regeln werden aus den zur Verfügung stehenden Trainingsdaten gebildet, wobei angenommen wird, dass die generierten Regeln auch außerhalb der in den Trainingsdaten vorhandenen Werte Gültigkeit besitzen. Es gibt zahlreiche Ursachen, durch die es möglich ist, diese Annahme der Allgemeingültigkeit (bzw. der Gültigkeit auf den Wertebereich der Betriebsdaten) zu verletzen, wie der bereits erwähnte Fehler, dass statt makroskopischer Zusammenhänge mikroskopische Beziehungen erlernt werden.[173] Hierdurch ist die Gültigkeit der erlernten Zusammenhänge lediglich für die Werte der zur Entwicklung genutzten Datensätze gegeben. Treten im späteren Betrieb Eingangsdaten auf, die außerhalb der Werte der dieser Datensätze liegen, ist es möglich, das Fehlverhalten auftritt, da die erlernten mikroskopischen Zusammenhänge ebenfalls, jedoch in einem veränderten makroskopischen Kontext, auftreten und hierdurch eine andere Ausgangsgröße besitzen als durch die mikroskopischen Zusammenhänge prädiziert. Im Fall des Leoparden-Beispiels eines Modells zur Identifikation

[170] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[171] Vgl. Winner, H.: Handbuch Fahrerassistenzsysteme (2015), S. 21.

[172] Vgl. Bergadano, F.: The Problem of Induction and Machine Learning (1991), S. 1073.

[173] Siehe Leopardenmuster-Beispiel in Unterkapitel 3.2.

von verschiedenen Objekten und Tieren aus Bilddateien hat das Modell die Fellstruktur als Merkmal zur Identifikation eines Leoparden erlernt und lässt dabei die Statur des Tieres zur Identifikation außer Betracht. Wird diesem Modell ein Sofa mit einer ähnlichen Fellstruktur gezeigt, prädiziert es mit hoher Zuverlässigkeit, dass es sich um einen Leoparden handelt.[174] Dadurch, dass diesem gelernten Modell durch die gute Prädiktionsrate auf dem Testdatensatz, welcher kein Objekt mit einem Leopardenmuster enthielt, eine hohe Leistungsfähigkeit zugeschrieben wird, stellt die Problematik der nicht vorhandenen Allgemeingültigkeit einen kritischen Fall in der Erbringung des Sicherheitsnachweises dar.[175] Die Eigenschaft, dass die Gültigkeit der Annahmen bzw. Regeln des gelernten Modells vollständig für den Einsatzbereich, d.h. insbesondere nicht nur für die in der Entwicklung genutzten Datensätze, ausreicht, wird im Folgenden mit dem Begriff Generalisierbarkeit beschrieben. Weitere, bereits bekannte ML-inhärente Fehlerursachen der Algorithmenarten, die in FAS eingesetzt werden, sind bei Faria[176] im Rahmen der Ausführungen zu dem von ihm entwickelten Sicherheitsnachweis explizit aufgelistet (siehe Abschnitt 3.2.2.2). Alle von Faria identifizierten Ursachen äußern sich ebenfalls in fehlender Generalisierbarkeit. Auch die durch Burton et al.[177] aufgelisteten Ursachen von funktionalem Fehlverhalten (siehe Abschnitt 3.2.2.1) resultieren in fehlender Generalisierbarkeit.

Zur Implementierung im Fahrzeug benötigen leistungsfähige gelernte Modelle, wie Tiefe Neuronale Netze, andere Hardware als die bisherigen konventionellen Algorithmen. Derzeit wird beispielsweise vor allem auf Grafikkarten für den Betrieb Tiefer Neuronaler Netze zurückgegriffen.[178] In diesem Bereich ist es möglich, dass neue hardwarespezifische Fehler auftreten, die mit der Nutzung der gelernten Algorithmen zusammenhängen. Diese Problematik wurde bereits von einem Grafikkartenhersteller aufgegriffen und als, in Einklang mit der ISO 26262, gelöst postuliert.[179] Daher wird kein Fokus auf diese Art der Fehler in der vorliegenden Betrachtung gelegt.

Salay et al.[180] stellte neben fehlender Generalisierbarkeit auch Gefahren durch eine veränderte Nutzer-System-Interaktion als ML-inhärente Fehler heraus, da es möglich ist, dass gelernte Modelle durch ihre hohe Leistungsfähigkeit und die hierdurch möglichen neuen Funktionalitäten als „intelligenter“ als konventionelle Algorithmen wahrgenommen werden. Hierdurch ist es möglich, dass Nutzer dem System zu sehr vertrauen, so dass sie beispielsweise bei ACC trotz Warnungen des Erreichens der Systemgrenzen keine Übernah-

[174] Vgl. Khurshudov, A.: Suddenly, a leopard print sofa appears (2015).

[175] Vgl. Koopman, P.; Wagner, M.: Autonomous Vehicle Safety (2017), S. 94.

[176] Faria, J. M.: Non-determinism and Failure Modes in Machine Learning (2017).

[177] Burton, S. et al.: Case for Safety of Machine Learning (2017).

[178] Vgl. DeAmbroggi, L.: Artificial intelligence in automotive (2016), S. 2ff.

[179] Vgl. nvidia: NVIDIA Announces World’s First Functionally Safe AI Self-Driving Platform (2018).

[180] Salay, R. et al.: An Analysis of ISO 26262 (2017).

me tätigen und hieraus ein Unfall resultiert. Da diese Wahrnehmung eines „intelligenten“ Systems und die hiervon abhängigen Gefahren und Fehler jedoch bereits durch die ISO/ PAS 21448 adressiert werden, werden diese aus der folgenden Betrachtung ausgeklammert.

Aufgrund dieser Analyse wird der Bereich der fehlenden Generalisierbarkeit weiter fokussiert, wodurch sich die folgende weitere Forschungsfrage ergibt: „*Wie ist es möglich, fehlender Generalisierbarkeit systematisch zu begegnen*?“. Für ein systematisches Vorgehen werden zunächst die Ursachen fehlender Generalisierbarkeit analysiert, um nach Ansätzen zu forschen, mit denen das Vorliegen der Ursachen zu identifizieren bzw. zu vermeiden ist. Daher lauten weitere, der obigen Frage untergeordnete Fragestellungen: „*Auf welche Ursachen ist fehlende Generalisierbarkeit zurückzuführen?*“ und „*Wie ist es möglich, diesen Ursachen strukturiert zu begegnen?*“ Diese werden jeweils einzeln in Kapitel 4 und 5 beantwortet.

4 Analyse der Generalisierbarkeit

Wie bereits in Unterkapitel 3.3 diskutiert, resultiert die Problematik der möglichen fehlenden Generalisierbarkeit aus der Nutzung des Induktionsprinzips zur automatisierten Generierung gelernter Modelle. Jedoch fehlt bisher eine systematische Auflistung konkreter Ursachen, die dazu führen, dass eine ausreichende oder mangelhafte Generalisierbarkeit für den Anwendungsfall des Modells erreicht wird. Um der Frage *„Auf welche Ursachen ist fehlende Generalisierbarkeit zurückzuführen?*“ nachzugehen, wird in Unterkapitel 4.1 eine Fehlerbaumanalyse durchgeführt. Für diese Fehler werden anschließend in Unterkapitel 4.2 mögliche Identifikations- und Vermeidungsmöglichkeiten analysiert und hinsichtlich dieser kategorisiert. Diese Kategorisierung wird in Unterkapitel 4.3 übersichtlich zusammengefasst.

4.1 Ableitung der Ursachen

Zur strukturierten Identifikation und Darstellung von Ursachen eignet sich die Methode der Fehlerbaumanalyse (englisch Fault Tree Analysis, kurz FTA). Ein Fehlerbaum stellt Kausalitäten in graphischer Form dar, wobei sich an der Spitze des „Baumes“ das sog. Hauptereignis oder Top Level Event (TLE) befindet. Dieses TLE stellt das unerwünschte Ereignis dar, dessen Ursachen zu untersuchen gilt. Es wird eine deduktive Analyse angewendet, bei welchem ausgehend vom Hauptereignis zunächst die Ursachen auf oberster Ebene identifiziert werden. Diese werden iterativ immer weiter untergliedert, bis die gewünschte Ursachengranularität erreicht wird. Zur besseren Beschreibbarkeit werden die einzelnen Ursachen mit Zahlen versehen. Diese befinden sich in rechteckigen Feldern, solange das Ende eines Astes und somit eine direkte Ursache noch nicht identifiziert wurde. Das Ende eines Astes wird mit einem Kreis gekennzeichnet.[181]

Im vorliegenden Fall stellt „fehlende Generalisierbarkeit“ das Hauptereignis dar. Die einzelnen Ursachen wurden anhand der Analyse des Entwicklungsprozesses gelernter Algorithmen (siehe Abschnitt 2.2.1) mit der Fragestellung *„An welchen Stellen ist es möglich, dass Ursachen fehlender Generalisierung auftreten?*“ in Kombination mit einer Literaturrecherche mit der Fragestellung „*Welche Ursachen treten in diesem Bereich auf?*“ identifiziert.

Zur Beantwortung der ersten Fragestellung sind die erste und zweite Ebene der durchgeführten FTA in Abbildung 4-1 dargestellt. Neben den eigentlichen Ursachen, die fehlende Generalisierbarkeit hervorrufen, wurden ebenfalls Ursachen analysiert, die verhindern,

[181] Vgl. Edler, F. et al.: Fehlerbaumanalyse in Theorie und Praxis (2015), S. 1f.

dass dieses Fehlverhalten bereits im Rahmen der Entwicklung auffällt. Da diese Ursachen einen Beitrag zur frühzeitigen Identifikation fehlender Generalisierung leisten, werden sie weiterhin betrachtet. Die direkten Ursachen fehlender Generalisierung finden sich im Trainings- und Validierungsprozess wider, wohingegen die Ursachen, die eine Detektion fehlender Generalisierung verhindern, im Testprozess verortet sind. Generell ist es möglich, dass Ursachen fehlender Generalisierung entweder in den Daten oder in den zur Modellgenerierung genutzten Algorithmen liegen. Da das Modell im Testprozess nicht mehr verändert wird, liegen die Ursachen zur fehlenden Detektion in diesen Abschnitten lediglich in den Daten. Im Validierungsprozess wird aus einer Menge an bereits trainierten Modellvarianten mit unterschiedlichen Modellparametern das am besten geeignete Modell anhand der Validierungsdaten ausgewählt, weshalb auch in diesem Entwicklungsschritt die Ursachen lediglich in den Daten identifiziert wurden.

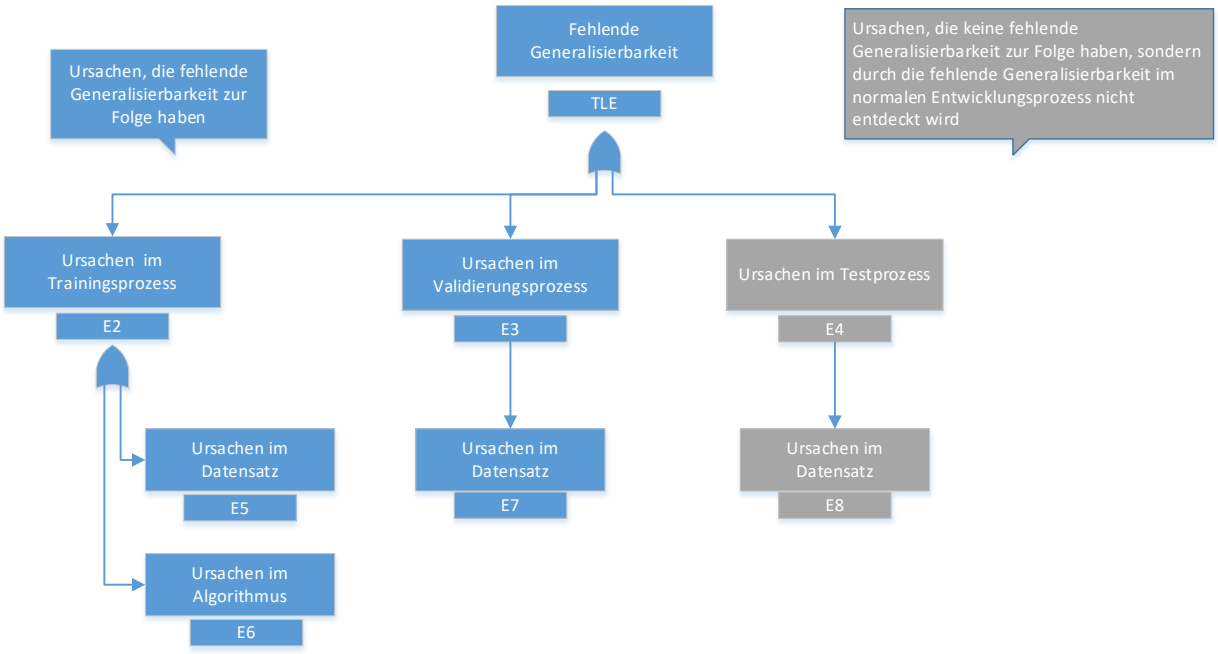


Abbildung 4-1: Fehlerbaum erste und zweite Ebene

Im Folgenden werden die identifizierten Ursachen innerhalb der einzelnen Unteräste separat in Abschnitten vorgestellt, wodurch in jedem Abschnitt der zweiten Frage nach den konkreten Ursachen in jedem Bereich nachgegangen wird. Es werden auch Ursachen aufgelistet, für die sich bereits Vermeidungsmaßnahmen etabliert haben, um ein umfassendes Bild zu ermöglichen. Auf die möglichen bzw. etablierten Vermeidungsmaßnahmen wird in Unterkapitel 4.2 und Kapitel 5 eingegangen.

4.1.1 E5: Ursachen im Datensatz (Trainingsprozess)

Der Ast E5, welcher die Ursachen beinhaltet, die innerhalb des Trainingsdatensatzes möglich sind, ist in Abbildung 4-2 abgebildet. Da aus den Trainingsdaten alle Zusammenhänge bzw. Gesetzmäßigkeiten für die Lösung der Problemstellung bzw. Aufgabe extrahiert wer-

den, besitzen diese einen maßgeblichen Einfluss auf die Generalisierbarkeit des Modells. Dabei lassen sich die Ursachen fehlender Generalisierbarkeit in eine zu geringe Qualität oder in eine unzureichende Quantität der Trainingsdaten gruppieren.

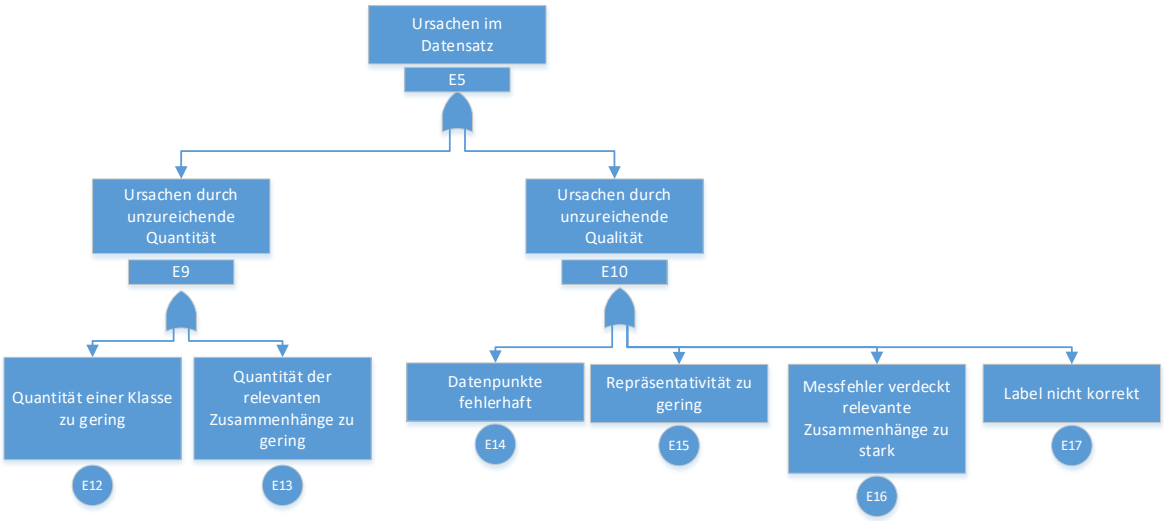


Abbildung 4-2: Ursachen im Trainingsdatensatz

E12: Quantität einer Klasse zu gering

Die Unterrepräsentanz einer Klasse im Rahmen einer Klassifikation[182] stellt eine mögliche Ursache für fehlende Generalisierbarkeit dar. Diese Problematik ist unter dem Begriff „imbalanced data“ oder „skewed classes“ bekannt und bezieht sich nur auf Supervised Lernansätze.

Die Verbesserung des Klassifikationsergebnisses des gelernten Modells während des Trainingsprozesses wird normalerweise durch die Berechnung der Vorhersagegenauigkeit mittels einer Konfusionsmatix durchgeführt. Die Spalten der Matrix stellen die vorhergesagte Klasse, die Zeilen die tatsächliche Klasse dar. Richtig bzw. korrekt als negativ klassifizierte Beispiele sind bei *TN* („true negative“) eingetragen, *FP* stellt die Anzahl der inkorrekt als positiv eingestuften Beispiele („false positives“) dar. Mit *FN* wird die Anzahl der falsch als negativ eingestuften Beispiele („false negatives“) und *TP* die Anzahl der korrekt eingestuften positiven Beispiele („true positives“) bezeichnet. Die Korrektklassifikationsrate (*Acc*)[183] wird wie folgt berechnet:[184]

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{4.1}$$

[182] Siehe Abschnitt 2.2.2.

[183] Englisch: Accuracy.

[184] Vgl. Chawla, N. V.: Data Mining for Imbalanced Datasets (2010), S. 876.

Besteht die Aufgabe beispielsweise darin, vorherzusagen, ob sich in den nächsten zwei Sekunden ein Unfall mit Todesfolge abhängig von Ego-Fahrdynamikgrößen und Bewegungsgrößen umgebender Verkehrsteilnehmer ereignet, so liegt typischerweise aufgrund des seltenen Auftretens eines Unfalls im Vergleich zu gefahrenen Kilometern[185] ein Datensatz mit vielen unkritischen Situationen (= „negatives" Ereignis) und einigen wenigen kritischen Situationen vor, bei denen noch weniger in einem tatsächlichen Unfall (= „positives" Ereignis) resultierten. Durch eine solche Klassenverteilung existiert nur eine relativ gesehen sehr kleine Menge an Daten, die überhaupt ein *TP* hervorrufen könnten.[186] Das führt dazu, dass die Vorhersagegenauigkeit sehr hoch ausfällt, auch wenn das gelernte Modell immer das Vorliegen einer unkritischen Situation vorhersagt, unabhängig der tatsächlich vorliegenden Situation.

E13: Quantität der relevanten Zusammenhänge zu gering

Im Bereich der Quantität ist es möglich, dass die in den Trainingsdaten vorhandene Gesamtmenge der für die Problemlösung benötigten relevanten Zusammenhänge für die Komplexität der Problemstellung nicht ausreicht. Neben der Komplexität der Problemstellung spielt auch die Komplexität des verwendeten Algorithmus eine wichtige Rolle. Komplexere Algorithmen benötigen eine höhere Anzahl an relevanten Zusammenhängen als einfachere Lernverfahren.[187]

E14: Datenpunkte fehlerhaft

Sind Datenpunkte fehlerhaft, d.h. liegen beispielsweise Ausreißer vor oder wurden korrupte Daten aufgenommen, so sind die hieraus erlernten Zusammenhänge nicht in der Realität vorhanden. In diese Ursache werden Label bzw. der Labelprozess nicht einbezogen, da dieser Aspekt in einer eigenen Ursache Rechnung getragen wird.

E15: Repräsentativität zu gering

Neben einer ausreichenden Menge an Trainingsdaten ist es für eine ausreichende Generalisierbarkeit ebenfalls notwendig, dass diese Menge eine genügend hohe Repräsentativität der späteren Betriebsbedingungen besitzt.[188] Besteht die Aufgabenstellung im späteren Betrieb z.B. in einem Clustering von Fahrstilen und enthält der Trainingsdatensatz nur zwei der drei Fahrstile, die in der Realität vorkommen, wird der darauf angewendete Clusteringalgorithmus die geforderte Trennung in zwei Cluster durchführen. Tritt der dritte

[185] Eine Abschätzung von Wachenfeld und Winner für das Jahr 2012 bezogen auf die Gesamtfahrleistung und die Zahl der tödlichen Unfälle in Deutschland berechnet den Abstand zwischen zwei tödlichen Unfällen zu 210 Millionen Kilometern (Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015), S.455).

[186] Vgl. Mukherjee, U.: How to handle Imbalanced Classification Problems (2017).

[187] Vgl. Brownlee, J.: How Much Training Data is Required for Machine Learning? (2017).

[188] Vgl. Burton, S. et al.: Case for Safety of Machine Learning (2017), S. 12ff.

Fahrstil, welcher nicht in den Trainingsdaten vorhanden ist, im Betrieb auf, wird der Algorithmus eine falsche Aussage treffen.

E16: Messfehler verdeckt relevante Zusammenhänge zu stark

Sind die zur Problemlösung erforderlichen Zusammenhänge im Datensatz vorhanden, ist es möglich, dass Messfehler diese für den Lernalgorithmus so stark verdecken, dass diese nicht in ihrer eigentlichen Form identifiziert werden und das gelernte Modell im Betrieb, wenn diese Messfehler nicht mehr oder anderweitig vorhanden sind, Fehlverhalten zeigt. Die Ursachen der Messfehler sind vielfältig und in der Literatur (beispielsweise Hering und Schönfelder[189]) ausführlich beschrieben.

E17: Label nicht korrekt

Eine unzureichende Qualität der Label führt zu inkorrekten gelernten Regeln des Modells im Vergleich zur Realität.[190] Diese Ursache tritt prinzipbedingt lediglich im Rahmen von Supervised-Lernansätzen auf. Sie ist dabei auf Label bezogen, die basierend auf den bereits aufgenommenen Ausgangsgrößen entweder manuell oder automatisiert nachträglich generiert wurden. Generelle unzureichende Qualität der aufgenommenen Daten (unabhängig ob Ein- oder Ausgangsgrößen) ist unter der Ursache E16 geführt und unterscheidet sich hinsichtlich späterer Vermeidungsmaßnahmen von der hier diskutierten Ursache. Nachträglich generierte Label finden dabei vor allem bei bildbasierten Ausgangsgrößen Anwendung, um beispielsweise Objekte wie Fußgänger anhand des zugehörigen Pixelbereichs manuell zu kennzeichnen. Dieser Datensatz wird z.B. genutzt, um ein Modell zur Fußgängerdetektion zu trainieren. Ein Beispiel für manuell generierte, nachträgliche Label außerhalb des Bildbereichs besteht in der Selbsteinschätzung eines Fahrers hinsichtlich seines Fahrstils basierend auf der vergangenen Fahrt. Hier besteht das Problem darin, dass keine eindeutige Ground-Truth[191] vorhanden ist, auf die im Annotationsprozess Bezug genommen wird. Im obigen Beispiel ist es fragwürdig, auf welche Referenz die Fahrer sich selbst beziehen und wie sinnvoll dieses Label überhaupt für ein Training zu nutzen ist.

4.1.2 E6: Ursachen im Algorithmus (Trainingsprozess)

Neben den Trainingsdaten dienen die eingesetzten Algorithmen zur Generierung der modellinhärenten Regeln, weshalb auch diese Ursachen enthalten, die zu fehlender Generalisierung führen. Diese teilen sich in zwei Kategorien auf: Die „objective function“[192] und die tatsächliche Implementierung, dargestellt mit Abbildung 4-3.

---

[189] Hering, E.; Schönfelder, G.: Sensoren in Wissenschaft und Technik (2012).

[190] Vgl. Faria, J. M.: Non-determinism and Failure Modes in Machine Learning (2017), S: 313.

[191] sinngemäß Referenzwahrheit oder zugrunde liegende Wahrheit

[192] Siehe Abschnitt 2.2.1.

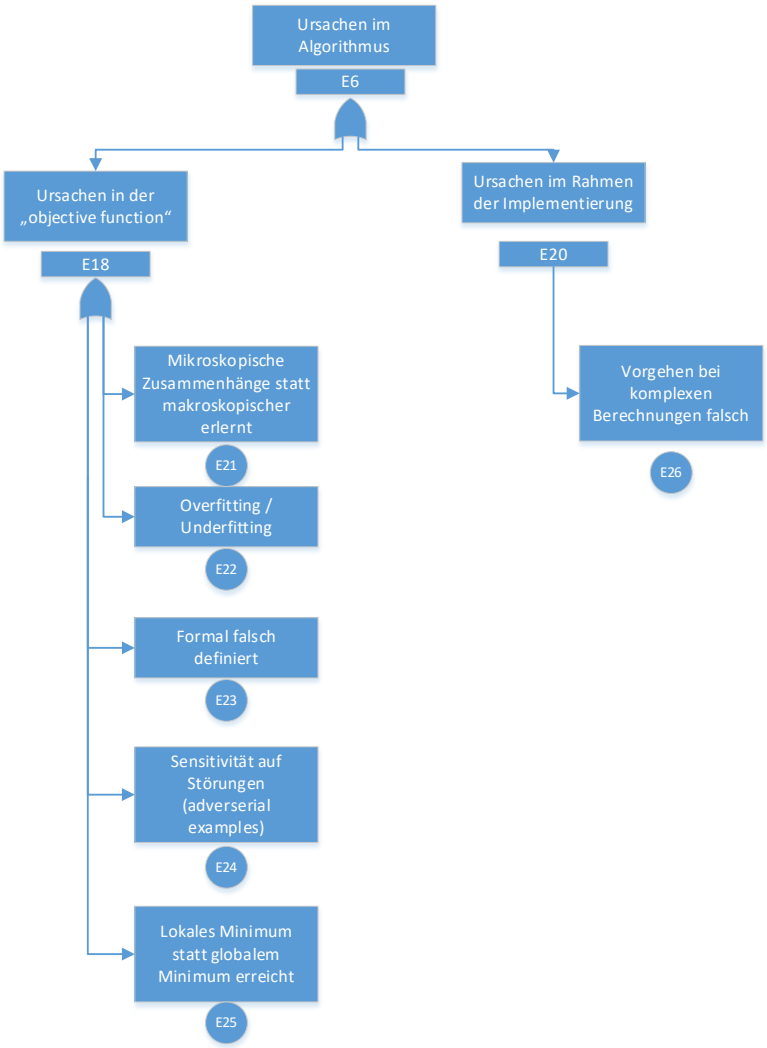


Abbildung 4-3: Ursachen in den Trainingsalgorithmen

E21: Mikroskopische anstelle makroskopischer Zusammenhänge erlernt

Eine Ursache für unzureichende Generalisierung besteht darin, dass die durch die Zielfunktion generierten Zusammenhänge aus mikroskopischen anstelle von makroskopischen Beziehungen der Daten bestehen. Die Ursache ist bereits in Unterkapitel 3.3 anhand des Beispiels mit der Fellstruktur eines Tieres als mikroskopisches Merkmal erläutert. Sie tritt bei komplexen Algorithmen auf, welche sich auf lokale Bereiche der Daten fokussieren.[193] Ein weiteres Beispiel besteht in einem Neuronalen Netz, das zur bildbasierten Unterscheidung genutzt wird, ob eine Person Lippenstift trägt oder nicht. Das Netz sagt dabei fälschlicherweise das Vorhandensein von Lippenstift als wahrscheinlich voraus, selbst wenn der Mund mit einem schwarzen Balken verdeckt ist. Das liegt daran, dass das Netz einen Zusammenhang zwischen Augen-Make-Up und Lippenstift gelernt hat, anstatt lediglich den Fokus auf den Lippenstift an sich zu setzen.[194]

[193] Vgl. Khurshudov, A.: Suddenly, a leopard print sofa appears (2015).

[194] Vgl. Zhang, Q.-s.; Zhu, S.-c.: Visual interpretability for Deep Learning (2018), S. 3.

E22: Overfitting/ Underfitting

Besitzt der gewählte Lernalgorithmus mehr Anpassungsparameter als vom zur Verfügung stehenden Datensatz gerechtfertigt ist, ist eine Überanpassung bzw. Overfitting möglich, wie in Abbildung 4-4 dargestellt.[195a] Durch diese Überanpassung an den Datensatz ist die erzielte Leistungsfähigkeit auf dem Trainingsdatensatz hoch. Im Gegensatz hierzu führen zu wenige Anpassungsparameter zu Underfitting.[195b] Underfitting wird jedoch im Entwicklungsprozess normalerweise durch eine geringe Leistungsfähigkeit bereits im Trainingsprozess schnell identifiziert. Overfitting ist mit dem Validierungsdatensatz identifizierbar, worauf in Kapitel 4.2 eingegangen wird.

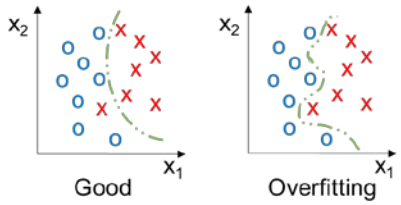


Abbildung 4-4: Overfitting

E23: Formal falsch definiert

Formale Fehler in der Aufstellung der Funktion, wie beispielsweise Vorzeichenfehler, stellen eine Ursache fehlender Generalisierbarkeit dar. Normalerweise wird diese Ursache durch die resultierende geringe Leistungsfähigkeit des gelernten Modells bei Supervised und Unsupervised Ansätzen bereits im Trainingsprozess offenbart.

E24: Sensitivität auf Störungen (adverserial examples)

Die unter dem Begriff „adverserial examples“ bekannt gewordene Problematik der Störungssensitivität stellt ebenfalls eine Ursache fehlender Generalisierbarkeit dar. Durch eine Generierung von Störungsgrößen, die auf die größtmögliche Änderung der Ausgangsgröße bei gleichzeitig kleinstmöglicher Änderung der Eingangsgröße optimiert sind, werden Vorhersagen des gelernten Modells hervorgerufen, die trotz hoher Selbstbewertung der Zuverlässigkeit der Vorhersage falsch sind. Das menschliche Detektionsvermögen wird von diesen Störgrößen nicht beeinflusst, was im Rahmen einer Objektdetektion im Bildbereich deutlich wird.[196] [197] Wie in Abbildung 4-5 gezeigt, wird ein originales Bild (links) mit einer Störung (Mitte) beaufschlagt, wodurch ein verändertes Bild entsteht (rechts). Für Menschen stellt auch das Bild mit Störgrößen eindeutig einen Panda dar, wohingegen ein gelerntes Neuronales Netz mit hoher Zuverlässigkeit das Vorliegen eines Gibbons vorhersagt.

---

[195] Vgl. Everitt, B.; Skrondal, A.: The Cambridge dictionary of statistics (2010), a: S. 318; b: S. 440.

[196] Vgl. Szegedy, C. et al.: Intriguing properties of neural networks (2013).

[197] Vgl. Goodfellow, I. J. et al.: Explaining and harnessing adversarial examples (2014).

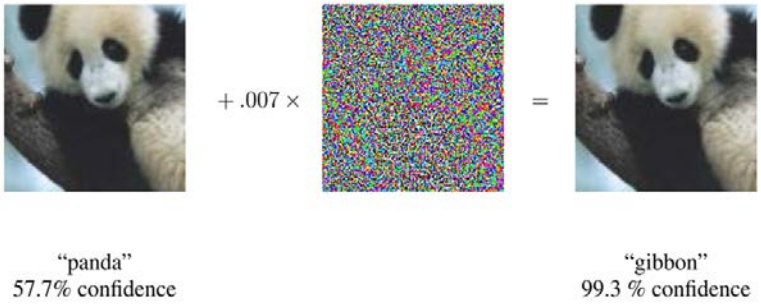


Abbildung 4-5: Adverserial example[198]

E25: Lokales statt globalem Minimum erreicht

Im Rahmen der Suche des Optimums der „objective function" ist es möglich, dass lokale Optima statt dem globalen Optimum erreicht werden und hierdurch das resultierende Modell nicht die bestmögliche Lösung darstellt.[199] Fehlende Generalisierbarkeit resultiert in diesem Fall aus der möglichen besseren Darstellung der Zusammenhänge der Trainingsdaten, deren Suche lediglich zu früh abgebrochen wurde. Diese Ursache tritt nicht im Rahmen von konvexen Funktionen, wie beispielsweise SVM, auf.[200]

E26: Vorgehen bei komplexen Berechnungen falsch

Beim Training des Modells sind teilweise komplexe Berechnungen notwendig. Für eine effiziente Implementierung wird häufig der Vektorraum genutzt, wodurch verschiedene Matrixoperationen, wie die Berechnung der Matrixinversen, benötigt werden.[201] Je nach verwendeter Implementierungssprache werden diese Berechnungen unterschiedlich, z.T. mit Vereinfachungen durchgeführt, wodurch ebenfalls fehlende Generalisierbarkeit möglich ist.

4.1.3 E3: Ursachen im Validierungsprozess

Im Ast E3 (siehe Abbildung 4-6) sind die Ursachen fehlender Generalisierbarkeit, die auf den Validierungsprozess zurückzuführen sind, aufgelistet. Da der Validierungsdatensatz zur Optimierung bzw. zur Festlegung von Hyperparametern[202] des gelernten Modells genutzt wird, ist dieser für die Gestaltung des finalen gelernten Modells relevant.[203] Das bedeutet, dass auch in diesem Entwicklungsschritt Anpassungen vorgenommen werden, die die im Modell enthaltenen Annahmen beeinflussen.

---

[198] Nach Goodfellow, I. J. et al.: Explaining and harnessing adversarial examples (2014), S.3.

[199] Vgl. Taylor, B. J. et al.: Verification and validation of neural networks (2003).

[200] Vgl. Burges, C. J.; Crisp, D. J.: Uniqueness of the SVM solution (2000), S. 223.

[201] Vgl. Faria, J. M.: Non-determinism and Failure Modes in Machine Learning (2017), S. 315.

[202] Siehe Abschnitt 2.2.1.

[203] Vgl. Bishop, C. M.: Pattern recognition and machine learning (2006), S. 32f.

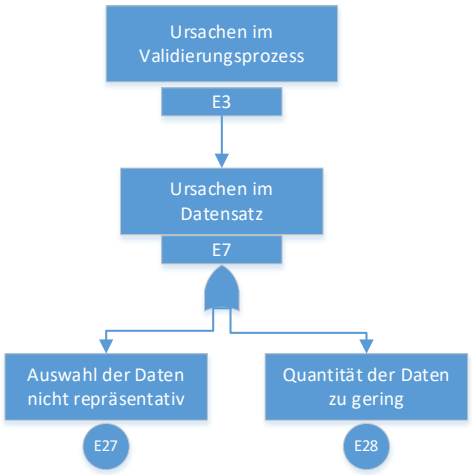


Abbildung 4-6: Ursachen im Validierungsprozess

E27: Auswahl der Daten nicht repräsentativ

Es ist möglich, dass eine fehlerhafte Optimierung des gelernten Modells in nicht-repräsentativen Daten hinsichtlich des Betriebsbereichs begründet liegt. Wird bei einer Regression im Validierungsprozess beispielsweise entschieden, welche Ordnung das final verwendete Polynom besitzt, kann eine nicht-repräsentative Auswahl an Validierungsdaten dazu führen, dass sich für ein zu niedriges oder hohes Polynom entschieden wird, was in einer fehlerhaften Modellannahme resultiert. Dies äußert sich normalerweise in einer geringen Leistungsfähigkeit im Testdatensatz bei Supervised-Ansätzen oder sonstiger Leistungsevaluation[204] bei Unsupervised-Ansätzen.

E28: Quantität der Daten zu gering

Werden im Vergleich zur Aufgabenkomplexität und den für den Algorithmus benötigten Hyperparametern zu wenige Daten zur Validierung genutzt, sind die hiermit getroffenen Parameteranpassungen ggf. nicht der Realität entsprechend. Dies liegt ebenfalls in der fehlenden Repräsentativität dieser zu geringen Menge begründet. Um eine ausreichende Repräsentativität der Daten zu erhalten, wird neben der reinen Repräsentativität der Datenauswahl (siehe Ursache E28) auch eine genügend große Menge an Daten benötigt, damit diese Repräsentativität dargestellt werden kann.

4.1.4 E4: Ursachen im Testprozess

Wie bereits erwähnt, führen die in diesem Abschnitt gelisteten Ursachen nicht zu fehlender Generalisierbarkeit, sondern verhindern eine Identifikation jener. Es ist theoretisch möglich, dass fehlende Generalisierbarkeit vollständig im Testprozess entdeckt wird, wenn die Testdaten alle im Betrieb vorkommenden Situationen abdecken. Hierdurch würde auch die

[204] Z.B. Evaluation der verbleibenden Abstände der Datenpunkte zu Clusterschwerpunkten bei K-Means-Clustering (siehe Abschnitt 6.2.1)

Sicherheit eines Systems beweisbar werden. Allerdings ist dies aufgrund der notwendigen, enorm hohen Anzahl an Testdaten, der noch höheren Anzahl an Trainingsdaten[205], sowie dem Aufwand zur Erfassung dieser nicht realistisch. Daher bestehen die Ursachen, weshalb fehlende Generalisierbarkeit im Testprozess nicht identifiziert wird, in einer zu geringen Quantität der Daten (E29) und in einer nicht-repräsentativen Auswahl der Daten (E30) (siehe Abbildung 4-7).

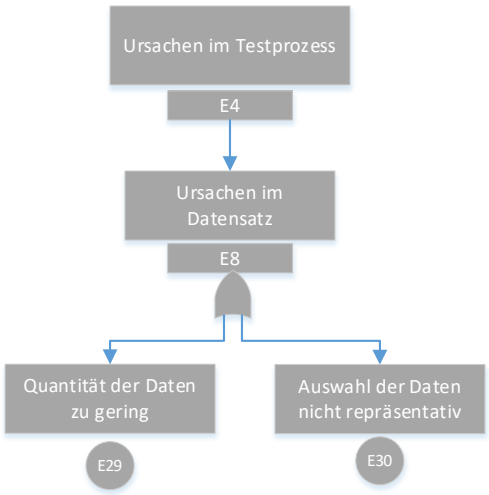


Abbildung 4-7: Ursachen, die Identifikation fehlender Generalisierbarkeit verhindern

4.1.5 Zusammenfassung

Die meisten Ursachen für fehlende Generalisierbarkeit werden im Trainingsprozess identifiziert. Das liegt daran, dass hier die grundlegenden Annahmen des gelernten Modells getroffen werden. Der vorgestellte Fehlerbaum erhebt nicht den Anspruch auf Vollständigkeit, sondern dient als erste Basis, dem Problem der fehlenden Generalisierbarkeit strukturiert zu begegnen. Deshalb wird die Frage „*Welche Ursachen treten in diesem Bereich auf?*“ nicht abschließend beantwortet. Falls weitere Ursachen identifiziert werden, ist es möglich, diese in die bestehende Struktur zu integrieren.

4.2 Gliederung der Ursachen

Während der Untersuchung jeder einzelnen Ursache fehlender Generalisierbarkeit aus Unterkapitel 4.1 hinsichtlich möglicher Vermeidungs- und Identifikationsmaßnahmen wurde festgestellt, dass die Ursachen sich hinsichtlich dieser Maßnahmen in drei Kategorien gliedern lassen:

[205] Der größte Teil der zur Verfügung stehenden Daten wird für das Training eines Modells genutzt, lediglich ca. 20-30% der Daten werden zum Testen verwendet (Vgl. Bronshtein, A.: Train/Test Split and Cross Validation in Phyton (2017)).

- Kategorie A: Ursachen, die sich im Entwicklungsprozess vermeiden lassen
- Kategorie B: Ursachen, deren Vorliegen direkt überprüfbar ist
- Kategorie C: Ursachen, deren Vorliegen nicht direkt überprüfbar ist

Im Folgenden wird die Zuordnung der einzelnen Ursachen (E12 - E28) analysiert.

E12: Quantität einer Klasse zu gering

Die Ursache E12 (Quantität einer Klasse zu gering) lässt sich durch Methoden der Datenvisualisierung wie eine Hauptkomponentenanalyse bzw. Principal Component Analysis[206] im Vorfeld des Lernprozesses vermeiden. Mit diesen Methoden wird ein hochdimensionaler Datenraum niedrigdimensional, d.h. menschlich interpretierbar, dargestellt. Hierdurch lässt sich diese Ursache zur **Kategorie A** zählen. Ein Überblick bzw. eine Einführung in die Techniken zur Visualisierung von hochdimensionalen Daten findet sich bei Viégas und Wattenberg[207]. Allerdings stellt eine solche Datenvisualisierung keine Methode dar, mit der die Ursache sicher vermieden wird, sondern lediglich die Wahrscheinlichkeit des Auftretens der Ursache mindert. Dies liegt u.a. daran, dass durch die niedrigdimensionale Darstellung Zusammenhänge der originalen Darstellung nicht komplett wiedergegeben werden. Hierdurch ist diese Ursache **zusätzlich in Kategorie C** einzuordnen. Werden die starken Klassengrößenunterschiede zwar erkannt, aber lassen sich prinzipbedingt nicht vermeiden (wie im Unfalldatenbeispiel), existieren verschiedene Lösungsansätze. Durch diese ist es möglich, mit den Klassengrößenunterschieden umzugehen, ohne die kleine Klasse im gelernten Modell zu vernachlässigen.[208] [209] [210]

E13: Quantität der relevanten Zusammenhänge zu gering

Da die benötigte Quantität an Daten bzw. den darin enthaltenen Zusammenhängen für maschinelles Lernen ein offenes Forschungsgebiet darstellt, existieren bisher nur sehr spezielle Abschätzungen, wie groß ein Datensatz für ein bestimmtes Lernverfahren bzw. eine bestimmte Anwendung mindestens zu sein hat.[211] [212] Metriken oder Methoden, die die Datenquantität bzw. die Quantität der relevanten Zusammenhänge bewerten, wurden nicht identifiziert. Hierdurch verbleibt zur Vermeidung bzw. Identifikation einer zu geringen Anzahl an relevanten Zusammenhängen in den Trainingsdaten die niedrigdimensionale Visualisierung[213] oder Zerlegung der Daten bzw. deren eingehende Analyse. Die Eingrup-

[206] Vgl. Bishop, C. M.: Pattern recognition and machine learning (2006), S. 561ff.

[207] Viégas, F.; Wattenberg, M.: Visualization for Machine Learning (2018).

[208] Vgl. Japkowicz, N.: Learning from imbalanced data sets (2000).

[209] Vgl. Chawla, N. V.: Data Mining for Imbalanced Datasets (2010).

[210] Vgl. Mukherjee, U.: How to handle Imbalanced Classification Problems (2017).

[211] Vgl. Mukherjee, S. et al.: Estimating dataset size requirements (2003).

[212] Vgl. Figueroa, R. L. et al.: Predicting sample size required for classification performance (2012).

[213] Vgl. Viégas, F.; Wattenberg, M.: Visualization for Machine Learning (2018).

pierung der Ursache E13 erfolgt daher in **Kategorie A** und **zusätzlich in Kategorie C**, da auch bei der Durchführung einer Datenanalyse durch die fehlende Metrik keine eindeutige Aussage hinsichtlich des Ausreichens der vorliegenden Zusammenhänge möglich ist.

E14: Datenpunkte fehlerhaft

Neben einer Datenvisualisierung[213] zur Detektion von Ausreißern oder sonstigen Auffälligkeiten wurde ein Ansatz von Chakarov et al.[214] identifiziert, in welchem bei Falschklassifikationen bei Klassifikationsaufgaben automatisiert deren Ursache im Datensatz analysiert wird. Da ein solches Vorgehen allerdings nur für Klassifikationen gefunden wurde und die visuelle Detektion keinen Ausschluss von fehlerhaften Datenpunkten garantiert, ist die Ursache E14 sowohl in **Kategorie A als auch in B und C** einzuordnen.

E15: Repräsentativität zu gering

Eine ausreichende Repräsentativität festzustellen bzw. fehlende Repräsentativität zu vermeiden, gestaltet sich schwierig, da hierzu ein vollständiges Wissen über die Problemstellung und die im Betrieb auftretenden Eingangsdaten notwendig ist. Nur hierdurch lässt sich ableiten, ob eine Untermenge dieser Daten als Trainingsdaten repräsentativ für die Gesamtmenge ist. Natürlich sind ggf. große Lücken, wie eine gesamte Klasse, die in den Daten fehlt, über eine Analyse der Trainingsdaten[213] identifizierbar, doch auch in diesem Fall ist fehlende Repräsentativität nicht komplett auszuschließen. Salay und Czarnecki[215] empfehlen die Aufstellung von Anforderungen an den Datensatz, wie die Erfüllung bestimmter Abdeckungsraten, die vor dessen Verwendung zu überprüfen sind. Konkrete Beispiele wurden hierbei jedoch nicht genannt. Hierdurch erfolgt die Gliederung dieser Ursache in **Kategorie A und Kategorie C**.

E16: Messfehler verdeckt relevante Zusammenhänge zu stark

Das Vorliegen von Messfehlern, wie beispielsweise Rauschen, ist durch Datenanalyse[213] im Vorfeld der Entwicklung des gelernten Modells identifizierbar. Allerdings bleibt hierbei die Frage offen, ob die Stärke der vorliegenden Messfehler ausreicht, damit das die relevanten Zusammenhänge für den Lernalgorithmus verdeckt oder der Algorithmus hierdurch Artefakte erlernt, die im Betrieb nicht in dieser Form auftreten werden. Es folgt daher eine Zuordnung zu **Kategorie A und zusätzlich C**.

E17: Label nicht korrekt

Es handelt sich bei E17 um eine Ursache aus der **Kategorie A**. Eine mögliche Vermeidungsmaßnahme stellt die mehrfache Überprüfung der Label durch unterschiedliche Methoden oder Personen dar. Dabei ist es möglich, lediglich eine Stichprobe der Label zu überprüfen und hierauf basierend eine Aussage zur Signifikanz der Überprüfung zu liefern oder alle Label bspw. visuell zu überprüfen. Allerdings verschafft diese Überprüfung nur

[214] Vgl. Chakarov, A. et al.: Debugging Machine Learning Tasks (2016).

[215] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018), S. 21.

Abhilfe, wenn eine eindeutige Ground-Truth vorhanden ist, auf die im Annotationsprozess Bezug genommen wird. Liegen beispielsweise subjektive Einschätzungen, wie die Evaluation des eigenen Fahrstils, als Label vor, besteht die Vermeidungsmaßnahme dieser Ursache eher im Verwerfen des Labels und ggf. Umstieg auf einen Unsupervised-Lernansatz, da diesen Labeln aufgrund der Subjektivität nicht zu vertrauen ist.

E21: Mikroskopische Zusammenhänge statt makroskopischer erlernt

Um die Art der Zusammenhänge zu überprüfen, die zur Entscheidung vom gelernten Modell herangezogen werden, ist eine Extraktion dieser Entscheidungen bzw. eine Verdeutlichung dieser Entscheidungen notwendig. Hierzu wurden bereits einige algorithmenspezifische Ansätze entwickelt, welche im Rahmen von Abschnitt 3.2.1.2 vorgestellt werden. Es handelt sich hierbei um niedrigdimensionale Visualisierungen der Vorgänge in einem Algorithmus, wobei vor allem Ansätze zur Visualisierung der Vorgänge in Neuronalen Netzen, die zur Klassifizierung von Objekten in Bildern genutzt werden, identifiziert wurden.[216] [217] [218] Hierdurch lässt sich beispielsweise der vorgestellte, fehlerhaft erlernte Zusammenhang zwischen Augen-Make-Up und Lippenstift im Rahmen der Klassifizierung, ob die Person im Bild Lippenstift trägt oder nicht, erkennen.[219] Wie in Abbildung 4-8 durch die hervorgehobenen Bildbereiche gezeigt, sind solche Visualisierungsmethoden in der Lage, die Bereiche der Eingangsdaten, die für eine bestimmte Vorhersage besonders interessant sind, zu kennzeichnen. Hierdurch ist es, zumindest im Bildbereich, möglich, fehlerhaft erlernte Zusammenhänge zu identifizieren, weshalb diese Ursache **Kategorie B** zuzuordnen ist. Allerdings sind die bisherigen Methoden nur für wenige Algorithmenarten geeignet, wodurch die Ursache **zusätzlich der Kategorie C** zugeordnet wird.

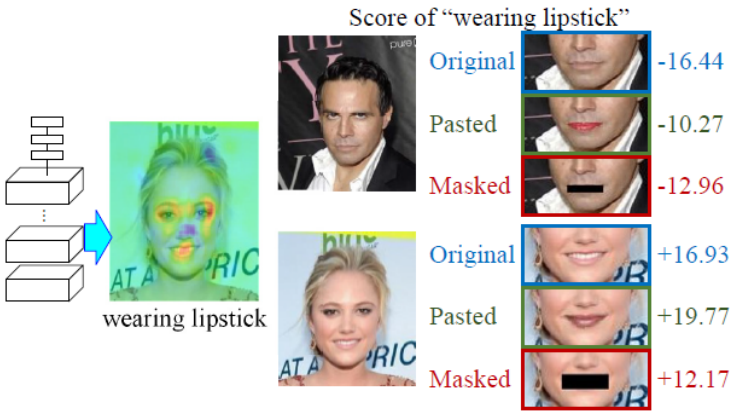


Abbildung 4-8: Visualisierung der für die Vorhersage relevanten Regionen[220]

[216] Vgl. Olah, C. et al.: Feature Visualization (2017).

[217] Vgl. Yosinski, J. et al.: Understanding Neural Networks (2015).

[218] Vgl. Zhang, Q.-s.; Zhu, S.-c.: Visual interpretability for Deep Learning (2018).

[219] Vgl. Zhang, Q. et al.: Examining CNN Representations with respect to Dataset Bias (2017).

[220] Zhang, Q. et al.: Examining CNN Representations with respect to Dataset Bias (2017).

E22: Overfitting/ Underfitting

Ein Beispiel für Ursachen, die zu **Kategorie B** zählen, findet sich in E22 (Overfitting/ Underfitting). Die Über- bzw. Unteranpassung des gelernten Modells ist nicht direkt zu vermeiden, jedoch lassen sich die beiden Fälle im Entwicklungsprozess identifizieren. Beide Fälle äußern sich bei Supervised-Ansätzen im Validierungsprozess, wenn die Fehler des gelernten Modells in Validierungs- und Trainingsdatensatz über eine steigender Anzahl an Parametern bzw. aufeinander aufbauenden Verbesserungsstufen des Modells miteinander verglichen werden (siehe Abbildung 4-9). Der Fehler des Trainingsdatensatzes sinkt dabei mit jeder Verbesserungsstufe, da sich das gelernte Modell immer stärker an den Trainingsdatensatz anpasst und immer weniger Fehler durch diese Anpassung möglich sind. Sinkt der Fehler des zur Validierung genutzten Datensatzes zunächst mit steigender Anzahl an Parametern und steigt danach wieder an, liegt ungefähr ab der Anzahl an Parametern bzw. ab der Verbesserungsstufe, an welcher der Fehler ansteigt, eine Überanpassung vor, da das gelernte Modell ab diesem Punkt die Validierungsdaten nichtmehr so korrekt vorhersagt, wie zuvor.[221] Im Bereich vor diesem Wendepunkt liegt wiederum Unteranpassung vor, da beide Fehler mit steigenden Parametern sinken.[222]

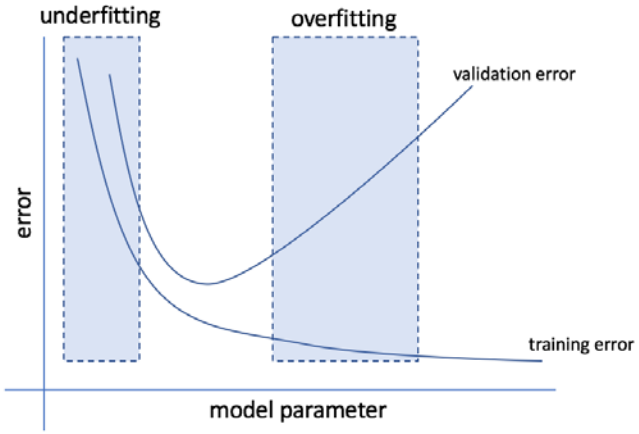


Abbildung 4-9: Identifikation von Überanpassung mittels des Validierungsdatensatzes[223]

Auch in Unsupervised-Ansätzen lässt sich Overfitting bzw. Underfitting identifizieren. Hierbei wird aufgrund der fehlenden Label nicht der Validierungsfehler des Modells zur Evaluation genutzt wird, sondern andere, für die jeweilige Problemstellung, passende Metriken. Eine Überanpassung bei Clustering-Algorithmen ist möglich, wenn mehr Cluster im Datensatz gefunden werden sollen als in der Realität vorhanden. Der Abstand zwischen den einzelnen Datenpunkten und den Cluster-Schwerpunkten sinkt dabei prinzipbedingt mit steigender Anzahl an Clustern. Allerdings existiert ein sog. „ellbow point“, bei dem der Abstand abrupt schwächer mit steigender Clusteranzahl abnimmt. Dieser Punkt wird als

[221] Vgl. Alpaydin, E.: Introduction to machine learning (2004), S. 77f.

[222] Vgl. Jordan, J.: Evaluating a machine learning model (2017).

[223] Jordan, J.: Evaluating a machine learning model (2017).

Referenz für eine angemessene Clusteranzahl empfohlen.[224] Aufgrund dessen wird beispielsweise die Steigung der durchschnittlichen Abstände zu den Clusterschwerpunkten betrachtet, um Überanpassung in diesem Fall zu vermeiden. Es sind jedoch auch andere Abstands- bzw. Fehlermaße möglich. Bei einer Clusteranzahl, die höher als der „ellbow point" (siehe Abbildung 4-10) liegt, ist es möglich, dass Überanpassung vorliegt.[225]

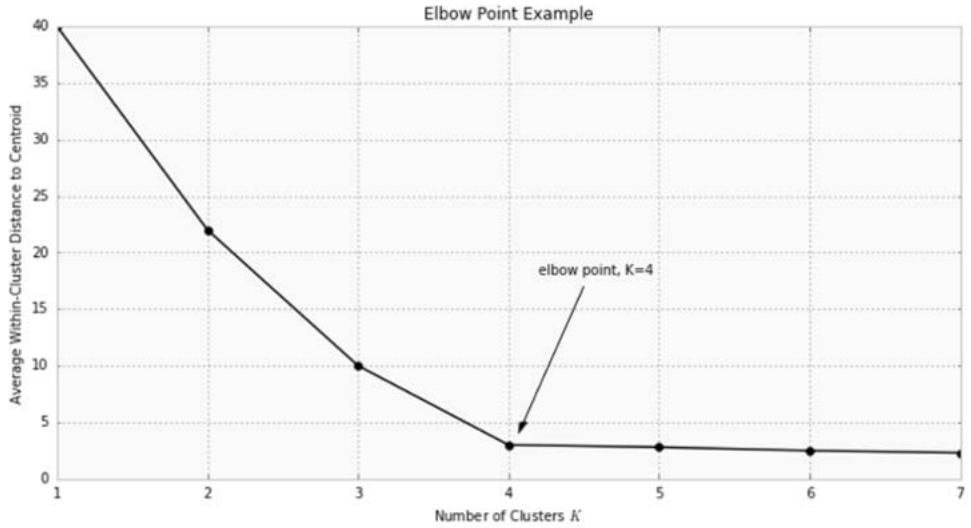


Abbildung 4-10: Ellbow Point zur Bestimmung einer angemessenen Clusterzahl[226]

Neben der direkten Überprüfung von Over- bzw. Underfitting existieren Vermeidungsmaßnahmen für Overfitting in Form von Regulierungsansätzen. Hierzu werden beispielsweise bei einer Regression hohe Polynom-Koeffizienten durch hohe „Kosten" bestraft.[227] Bei Neuronalen Netzen werden einzelne Neuronen während des Trainings „ausgeschaltet" (sog. Dropout), um Überanpassung zu vermeiden.[228] Aufgrund dieser Ansätze ist diese Ursache ebenfalls in **Kategorie A** einzuordnen.

Wie obige Erläuterungen allerdings zeigen, ist zwar eine direkte Überprüfung von Über- bzw. Unteranpassung ebenso wie die Verwendung von Ansätzen zur Vermeidung von Überanpassung möglich, allerdings verbleibt ein Bereich der Ungewissheit, wo genau Über- bzw. Unteranpassung auftritt und wo nicht. Hierdurch ist auch diese Ursache **zusätzlich der Kategorie C** zuzuordnen.

---

[224] Vgl. Mubaris: K-Means Clustering in Python (2017).

[225] Vgl. Trevino, A.: Introduction to K-means Clustering (2016).

[226] Trevino, A.: Introduction to K-means Clustering (2016).

[227] Vgl. Bishop, C. M.: Pattern recognition and machine learning (2006), S. 10.

[228] Vgl. Budhiraja, A.: Learning Less to Learn Better—Dropout in (Deep) Machine learning (2016).

E23: Formal falsch definiert

Wie bereits erwähnt, äußern sich formale Fehler in der Aufstellung des Algorithmus in einer geringen Leistungsfähigkeit des gelernten Modells. Zudem verhindert die Nutzung von vordefinierten Algorithmen aus bestehenden Bibliotheken wie scikit-learn[229] das Auftreten von formalen Fehlern. Hierdurch ist diese Ursache **Kategorie A und B** zuzuordnen.

E24: Sensitivität auf Störungen

Die Sensitivität des Algorithmus auf Störungen lässt sich direkt mit den vorgestellten „adverserial examples“[230] überprüfen. Hierzu geben Papernot et al.[231] einen Überblick an erforschten Sensitivitätsattacken inkl. zugehöriger Implementierung. Diese Ursache ist daher zu **Kategorie B** gehörig. Allerdings tritt diese Ursache in einigen Algorithmen, wie beispielsweise Neuronalen Netzen, zwangsweise auf, wodurch diese Überprüfung immer positiv ausfällt. Hier ist zu entscheiden, ob die hervorgerufene fehlende Generalisierbarkeit im Betrieb zu akzeptieren ist bzw. ob es realistisch ist, dass diese adverserial examples im Betrieb auftreten. Durch die Nutzung der Störbilder zum Training ist es möglich, die Sensitivität auf jene zu vermindern. Auch Methoden aus diesem Bereich wurden von Papernot et al.[231] zusammengefasst.

E25: Lokales Minimum statt globalem Minimum erreicht

Die Ursache, dass ein lokales statt dem globalen Optimum der *objective function* erreicht wurde, ist **Kategorie B** zuzuordnen. Durch mehrfache Initialisierung der Funktion und erneutem Training ist es möglich, zu untersuchen, ob und wie sich das Optimum verändert. Ändert sich das Optimum in eine „bessere“ Richtung, ist das zuvor gefundene Optimum nicht global. Ändert es sich nicht oder nur in eine „schlechtere“ Richtung, ist es ein Hinweis darauf, dass das zunächst gefundene Optimum global ist. Eine sichere Aussage hierzu ist jedoch nicht erreichbar. Es existieren in vorimplementierten Algorithmen bereits Funktionen, die eine mehrfache Initialisierung vornehmen und aus der resultierenden Menge an gelernten Modellen das „beste“ Optimum dem Nutzer als finales Modell präsentieren. Die Häufigkeit der Initialisierung innerhalb dieser Funktion wird vom Nutzer vorgegeben.[232] Hierdurch wird das Auftreten der Ursache bereits im Training verhindert, solange eine ausreichend große Menge an Initialisierungen vom Nutzer gewählt wird. Wenn solche vorimplementierten Funktionen genutzt werden, ist diese Ursache **zusätzlich in Kategorie A** einzusortieren. Generell besteht allerdings im Rahmen dieser Ursachenüberprüfung die Problematik, dass das Training von komplexen Algorithmen, wie Neuronale Netze, sehr rechenaufwändig ist, wodurch die Anzahl an Initialisierungen, die zur Überprüfung genutzt wird, begrenzt ist. Aufgrund dessen und da, wie bereits beschrieben, keine sichere Aussage

[229] https://scikit-learn.org.

[230] Vgl. Szegedy, C. et al.: Intriguing properties of neural networks (2013).

[231] Papernot, N. et al.: Technical Report on the CleverHans v2.1.0 Adversarial Examples Library (2016).

[232] Vgl. scikit-learn: sklearn.cluster.KMeans (2019).

möglich ist, ob das gefundene „beste“ Optimum auch tatsächlich das globale Optimum ist, wird diese Ursache **auch in Kategorie C** aufgeführt.

E26: Vorgehen bei komplexen Berechnungen falsch

Zur Verhinderung falscher bzw. zu stark vereinfachter Berechnungen innerhalb der Implementierung des Lernalgorithmus ist eine Kenntnis der Details der verwendeten Implementierungssprache notwendig. Durch die mögliche Verhinderung ist die Ursache in **Kategorie A** zu gliedern. Darüber hinaus ist es möglich, Unterschiede in Berechnungen über den Vergleich eines Algorithmus mit den gleichen Voraussetzungen (Hyperparametern etc.) in zwei unterschiedlichen Programmiersprachen zu erhalten, wodurch die Ursache nicht verhindert, jedoch identifiziert wird. Durch die Möglichkeit dieser aufwändigen Überprüfungsmethode ist die Ursache **zusätzlich in Kategorie B** zu führen.

E27: Auswahl der Daten nicht repräsentativ

Eine direkte Überprüfung der Ursache, dass die Daten, die zur Validierung herangezogen werden, nicht ausreichend repräsentativ sind, wurde nicht identifiziert. Allerdings existiert das Verfahren der k-fold-Cross-Validation, welches eine Methode darstellt, um alle Bereiche der Trainingsdaten einmal zur Validierung zu nutzen. Hierbei werden die Trainingsdaten in *k* einzelne Bereiche geteilt und jeweils eines der Datenteile als Validierungsdatensatz genutzt. Die übrigen *k*-1 Teile dienen zum Training des Algorithmus. Der Teil, der zur Validierung des gelernten Models genutzt wird, wird permutiert, wodurch *k* Trainingsläufe mit zugehöriger Validierung durchgeführt werden.[233] Hierdurch werden *k* unterschiedliche Ergebnisse aus der Validierung erreicht, die jeweils auf einem leicht veränderten gelernten Modell basieren. Diese Ergebnisse, beispielsweise eine Leistungsevaluation, werden normalerweise gemittelt. Abbildung 4-11 verdeutlicht dieses Vorgehen.

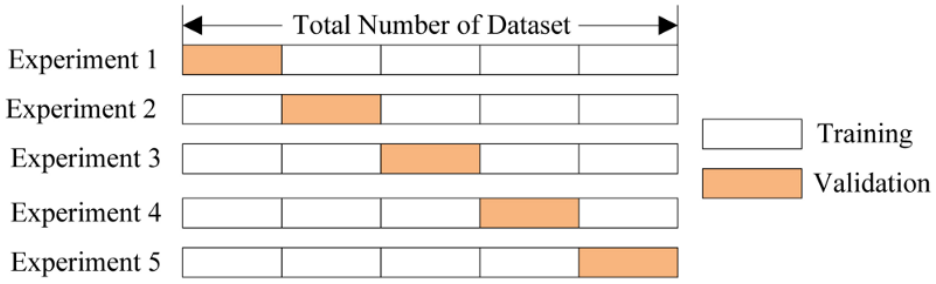


Abbildung 4-11: k-fold-Cross-Validation[234]

Es existieren neben der k-fold-Cross-Validation auch andere Methoden, die den Trainingsdatensatz intelligent zwischen Training und Validierung aufteilen, wie beispielsweise die Leave-one-out-Methode.[235] [236] Durch solche Methoden ist es zwar nicht möglich, die Ur-

[233] Vgl. Mitchell, T. M.: Machine learning (1997), S. 111f.

[234] Vgl. Mandava, s.: Cross Validation and HyperParameter Tuning in Python (2018).

[235] Vgl. Elisseeff, A.; Pontil, M.: Leave-one-out error (2003).

[236] Vgl. Evgeniou, T. et al.: Leave One Out Error (2004).

sache der zu geringen Repräsentativität zu vermeiden, allerdings wird durch die Permutation erreicht, dass das Mittel der Evaluation repräsentativ ist – vorausgesetzt, dass die Trainingsdaten eine ausreichende Repräsentativität der Betriebsumgebung enthalten. Hierdurch wird diese Ursache sowohl in **Kategorie A als auch in Kategorie C** eingeteilt.

E28: Quantität der Daten zu gering

Eine zu geringe Menge an Validierungsdaten lässt sich weder vermeiden, noch direkt überprüfen. Durch die Nutzung der k-fold-Cross-Validation (siehe E27) wird diesem Problem ebenfalls nicht begegnet, da die Menge an Daten für jeden Validierungsdurchgang gleichbleibt. Hierdurch ist diese Ursache in **Kategorie C** einzuordnen.

4.3 Überblick über die Kategoriezuordnung

Wie bereits in Unterkapitel 4.2 ausführlich erläutert, werden die Ursachen häufig mehr als einer Kategorie zugeordnet. Es existieren zwar teilweise Vermeidungsmaßnahmen, jedoch bieten diese nicht 100%-ige Sicherheit, dass die Ursache hierdurch tatsächlich nicht mehr auftritt. Tabelle 4-1 fasst die einzelne Kategoriezuordnung der Ursachen zusammen und dient als kurz gehaltene Antwort der Frage *„Auf welche Ursachen ist fehlende Generalisierbarkeit zurückzuführen?“* (siehe Unterkapitel 1.2).

Tabelle 4-1: Kategorie-Ursachen-Zuordnung

Kategorie A	• E12: Quantität einer Klasse zu gering • E13: Quantität der relevanten Zusammenhänge zu gering • E14: Datenpunkte fehlerhaft • E15: Repräsentativität zu gering • E16: Messfehler verdeckt relevante Zusammenhänge zu stark • E17: Label nicht korrekt • E22: Overfitting/ Underfitting • E23: Formal falsch definiert • E25: Lokales statt globalem Minimum erreicht • E26: Vorgehen bei komplexem Berechnungen falsch • E28: Auswahl der Daten nicht repräsentativ
Kategorie B	• E14: Datenpunkte fehlerhaft • E21: Mikroskopische anstelle makroskopischer Zusammenhänge erlernt • E22: Overfitting/ Underfitting • E23: Formal falsch definiert • E24: Sensitivität auf Störungen • E25: Lokales statt globalem Minimum erreicht • E26: Vorgehen bei komplexem Berechnungen falsch
Kategorie C	• E12: Quantität einer Klasse zu gering • E13: Quantität der relevanten Zusammenhänge zu gering • E14: Datenpunkte fehlerhaft • E15: Repräsentativität zu gering • E16: Messfehler verdeckt relevante Zusammenhänge zu stark • E21: Mikroskopische anstelle makroskopischer Zusammenhänge erlernt • E22: Overfitting/ Underfitting • E25: Lokales statt globalem Minimum erreicht • E27: Quantität der Daten zu gering • E28: Auswahl der Daten nicht repräsentativ

5 Überprüfung der Generalisierbarkeit

Nachdem in Kapitel 4 die Ursachen fehlender Generalisierbarkeit abgeleitet und diskutiert wurden, widmet sich Kapitel 5 der Fragestellung *„Wie ist es möglich, diesen Ursachen strukturiert zu begegnen?*“. In Unterkapitel 4.2 wurden bereits erste Ansätze zur möglichen Identifikation und Vermeidung der einzelnen Ursachen geliefert, wodurch die Ursachen in drei Kategorien hinsichtlich ihrer Identifikation bzw. Vermeidbarkeit eingeteilt werden. Es fehlt bisher jedoch an einem ganzheitlichen Ansatz, allen Ursachen zu begegnen, da Kategorie C Ursachen enthält, die weder vermeidbar noch direkt identifizierbar sind. Dieser Problematik wird sich in Unterkapitel 5.1 angenommen, in welchem ein strukturierter Ansatz zur Begegnung aller identifizierter und auch bisher nicht identifizierter Ursachen fehlender Generalisierbarkeit vorgestellt wird. Die nachfolgenden Unterkapitel behandeln anschließend die einzelnen Schritte bzw. Bestandteile dieses Ansatzes.

5.1 Ganzheitlicher Ansatz

Wie in Unterkapitel 4.2 und 4.3 gezeigt, lassen sich die Ursachen fehlender Generalisierbarkeit in drei Kategorien gliedern: Ursachen, die sich vermeiden lassen; Ursachen, die sich direkt überprüfen lassen und Ursachen, die sich weder vermeiden noch direkt überprüfen lassen. Für die ersten beiden Kategorien werden in Unterkapitel 4.2 bereits konkrete Vermeidungs- bzw. Überprüfungsmaßnahmen definiert. Problematisch gestaltet sich allerdings die Überprüfung der Ursachen, die in die dritte Kategorie (C) fallen. Eine Lösungsmöglichkeit besteht in der Überprüfung der Auswirkungen dieser Ursachen, d.h. der hieraus resultierenden fehlenden Generalisierbarkeit selbst. Der Vorteil an dieser Möglichkeit besteht darin, dass hierdurch nicht nur die Ursachen der Kategorie C überprüft werden, sondern ebenfalls gleichzeitig die Ursachen der anderen Kategorien. Falls beispielsweise eine Vermeidungsmaßnahme nicht korrekt implementiert wurde und eine Ursache trotz durchgeführter Maßnahme noch immer vorliegt, wird deren Auswirkung ebenfalls berücksichtigt. Zusätzlich werden auch Ursachen überprüft, die bisher noch nicht bekannt sind bzw. die im Rahmen des Fehlerbaums aus Unterkapitel 4.1 nicht identifiziert werden. Hierdurch ist ebenfalls die Problematik, dass dieser Fehlerbaum keinen Anspruch auf Vollständigkeit besitzt, gelöst.

Es ist möglich, durch die Überprüfung der Auswirkungen auf die anderen Methoden zur Vermeidung bzw. direkten Identifikation der Ursachen zu verzichten. Allerdings wird dies im vorgestellten Vorgehen nicht empfohlen, da der Aufwand in der Änderung des gelernten Modells aufgrund fehlender Generalisierbarkeit geringer ist, wenn die Ursachen früher im Entwicklungsprozess identifiziert werden. Abbildung 5-1 verdeutlicht das hieraus resultierende, stufenweise Vorgehen.

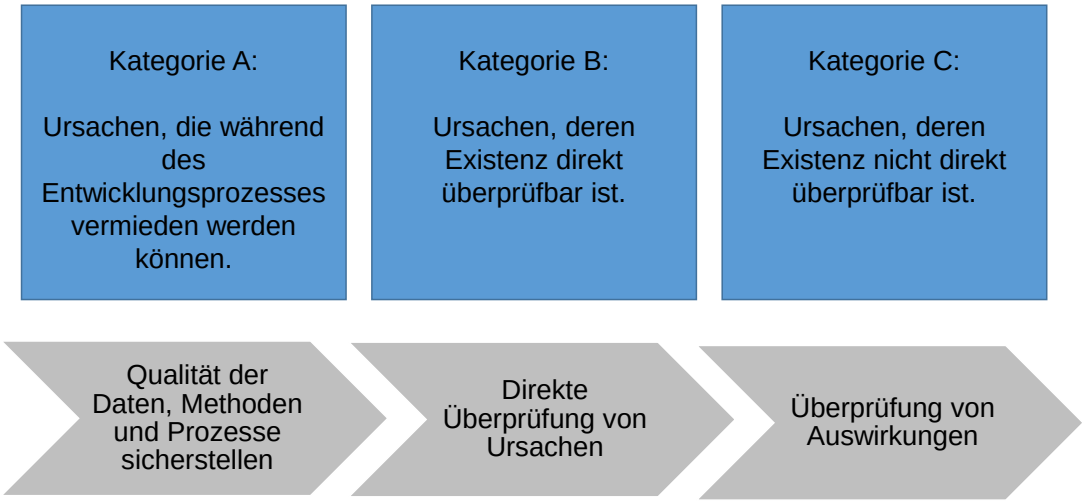


Abbildung 5-1: Gesamtansatz

Um die Auswirkungen der Ursachen bzw. die fehlende Generalisierbarkeit direkt zu überprüfen, werden zwei Methoden genutzt: Die Überprüfung von funktionalen Anforderungen auf der einen Seite und die Überprüfung von Anforderungen hinsichtlich der Robustheit des gelernten Modells auf der anderen Seite. Funktionale Anforderungen stellen Anforderungen an das Verhalten eines gelernten Modells unter festgelegten Bedingungen dar.[237] Ein Beispiel hierfür stellt im Rahmen einer Fußgängerdetektion die Anforderung dar, dass ein Objekt eine Ausdehnung von maximal 250 cm besitzt, um als Fußgänger klassifiziert zu werden. Durch ihren Funktionsbezug sind funktionale Anforderungen problem- bzw. aufgabenspezifisch. Im Rahmen einer konventionellen Programmierung werden diese Anforderungen im Vorfeld an das zu entwickelnde Modul gestellt und hierauf basierend die Implementierung durch den Programmierer vorgenommen. Anschließend wird die Erfüllung dieser Anforderungen durch das entwickelte Modul überprüft. So sieht es ebenfalls die anforderungsbasierte Vorgehensweise des V-Modells, welches von der ISO 26262 gefordert wird, vor. [238] Wie bereits diskutiert, besteht diese Notwendigkeit der Funktionsspezifikation im Rahmen gelernter Modelle nicht mehr, da die notwendige Funktionalität implizit durch die Trainingsdaten spezifiziert wird.[239] Für die Überprüfung der Generalisierbarkeit werden diese funktionalen Anforderungen dennoch benötigt und auch Salay und Czarnecki[240] fordern die Aufstellung dieser Anforderungen. Problematisch hierbei ist, dass ML vor allem dann eingesetzt wird, wenn Aufgabenstellungen nicht komplett spezifizierbar sind.[241] Zudem ist es möglich, dass die Aufstellung der funktionalen Anforderungen, auch wenn sie von Experten vorgenommen wird, unvollständig ist. Natürlich existiert das

[237] Vgl. Balzert, H.: Nichtfunktionale Anforderungen (2011), S. 109.

[238] Siehe Abschnitt 2.1.2.

[239] Siehe Unterkapitel 3.2.

[240] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018).

[241] Vgl. Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018), S. 7.

Problem der möglichen Unvollständigkeit der funktionalen Anforderungen auch bei bisherig eingesetzten konventionell programmierten Modellen. Allerdings wird zusätzlich zu deren Überprüfung ein gewachsenes Testkollektiv verwendet, um alle möglichen in der Realität vorkommenden Situationen zu überprüfen und hierdurch eine mögliche unvollständige Spezifikation zu kompensieren.[242] Da zum Training von gelernten Modellen möglichst viele, realitätsnahe Daten zu nutzen sind, ist es sinnvoll, dieses erwähnte Testkollektiv zum Training des Modells heranzuziehen. Hierdurch steht es jedoch nichtmehr zur Evaluation zur Verfügung. Selbst wenn dieses Testkollektiv nicht zum Training genutzt wird, enthält diese Testdatensammlung nicht die neuen Gefahren, die bei der Nutzung neuer Funktionalitäten, die durch gelernte Modelle möglich werden, auftreten (siehe schraffierte Fläche der Abbildung 5-2). Daher ist es nicht möglich, diesen bestehenden Ansatz zu nutzen, um einer möglichen unvollständigen Spezifikation zu begegnen.

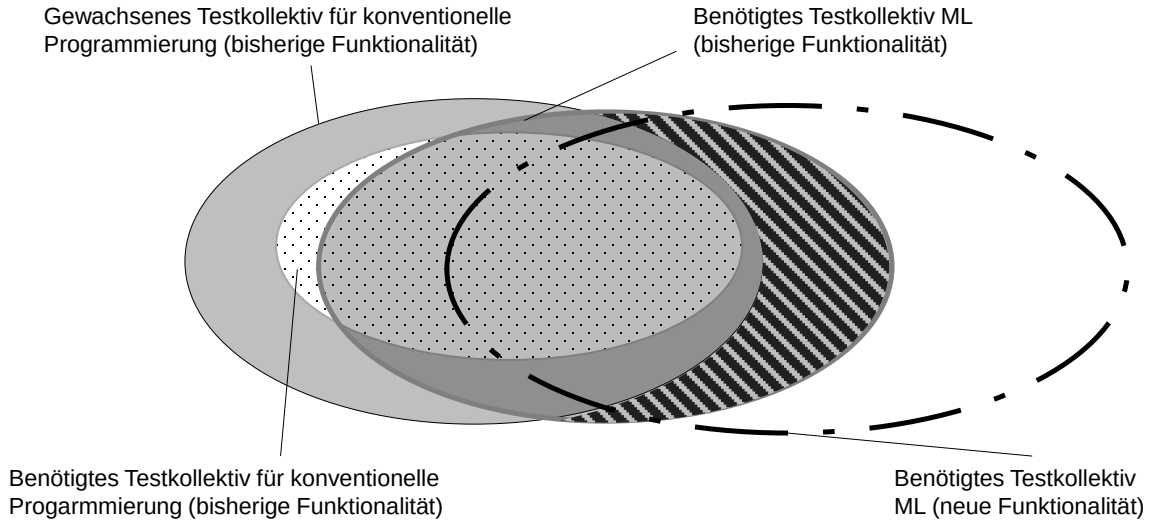


Abbildung 5-2: Benötigte Testkollektive im Vergleich

Unter anderem bedingt durch die vorgestellte Problematik der Unvollständigkeit funktionaler Anforderungen ist es notwendig, die Überprüfung der fehlenden Generalisierbarkeit durch die Überprüfung von Anforderungen hinsichtlich der Robustheit des gelernten Modells zu ergänzen.[243] Hierdurch werden Anforderungen, die implizit mit Generalisierbarkeit verbunden werden, explizit überprüft, um hierdurch eine umfassendere Überprüfung der Generalisierbarkeit zu erreichen als alleine durch das Testen der funktionalen Anforderungen. Bei konventionell programmierten Modellen ist eine solche Überprüfung nicht erforderlich, da die Generalisierbarkeit und deren implizite Anforderungen durch die explizite Umsetzung der funktionalen Anforderungen gegeben ist und zusätzlich ein gewach-

[242] Vgl. Singer, C.: Dissertation, Entwicklung von Testauswahlmethoden (2015).

[243] Weitere Gründe der Notwendigkeit für Robustheitsanforderungen werden in Unterabschnitt 5.4 vorgestellt.

senes Testkollektiv eingesetzt wird. Ein Beispiel einer solchen impliziten Anforderung stellt die korrekte Funktionalität eines gelernten Modells dar, auch wenn einzelne Datensätze aus dem Training entfernt wurden. Diese impliziten Anforderungen haben zum Ziel, die These der Generalisierbarkeit des gelernten Modells zu widerlegen, wodurch sie die Robustheit des Modells auf Änderungen fokussieren. Die zentrale Frage im Aufstellen dieser Robustheits-Anforderungen lautet daher „Wie ist es möglich, das gelernte Modell durch Testfälle oder Änderungen im Entwicklungsprozess so zu stören, so dass falsche Vorhersagen getroffen werden?“

Die Forschungsfrage „*Wie ist es möglich diesen Ursachen strukturiert zu begegnen?*“ (siehe Unterkapitel 1.2) wird daher wie folgt beantwortet:

Durch ein vierstufiges Vorgehen ist es möglich, den Ursachen fehlender Generalisierbarkeit strukturiert zu begegnen. Diese Stufen sind:

1. Sicherstellung der Qualität der Daten, Methoden und Prozesse
2. Direkte Überprüfung von Ursachen
3. Überprüfen von funktionalen Anforderungen
4. Überprüfen von Robustheitsanforderungen

Der Gesamtansatz trägt dazu bei, das Verhalten des gelernten Modells auch bei unbekannten Situationen besser einzuschätzen, da die Generalisierung die Extrapolation von Wissen auf unbekannte Bereiche darstellt. Hierdurch wird der Zielsetzung der ISO/ PAS 21448 nachgekommen, die in Abbildung 5-3 dargestellt ist.

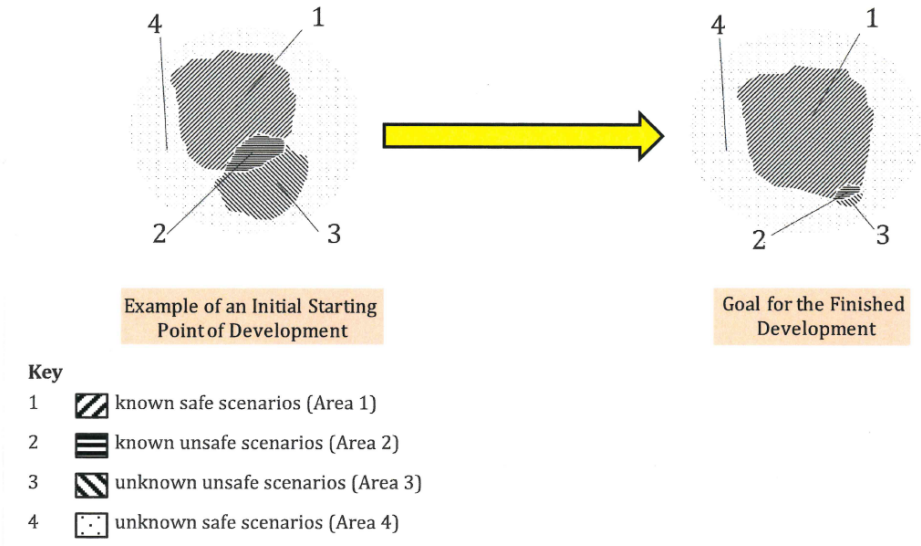


Abbildung 5-3: Reduktion der möglichen Fehler gemäß ISO/ PAS 21448[244]

[244] ISO: ISO/ PAS 21448 (2019), S. 8.

Durch die Überprüfung der funktionalen Anforderungen und deren Generalisierbarkeit in nicht durch Daten abgedeckte Bereiche wird bei Bestehen der funktionalen Anforderung der Bereich 4, der bekannten und sicheren Szenarien, vergrößert. Durch Aufdecken der Verletzung funktionaler Anforderungen und anschließender Verbesserung des Modells wird der Bereich 3, der unbekannten unsicheren Szenarien, verkleinert. Die Analyse der Robustheit des Modells vergrößert bei Bestehen der Robustheitsanforderung den Bereich 1, da beispielsweise durch die Veränderungen der Vorverarbeitung der Daten ein breiterer Bereich an Szenarien abgedeckt wird, in denen sicheres Verhalten vorliegt. Bei Verletzung der Robustheitsanforderungen werden Fälle des Bereich 3 aufgedeckt und durch die anschließende Verbesserung des Modells oder Einschränkung des Betriebsbereichs dieser Bereich verkleinert.

5.2 Qualität der Daten, Methoden und Prozesse

Der erste Schritt des Ansatzes besteht in der Begegnung der Ursachen, die in Kategorie A fallen und daher vermeidbar sind. Die Vermeidung beruht hierbei auf der Sicherstellung von ausreichender Qualität der eingesetzten Daten, Methoden und Prozesse durch Handlungsempfehlungen, die vor bzw. während der Entwicklung des gelernten Modells anzuwenden sind. Diese Empfehlungen werden bereits in Unterkapitel 4.2 vorgestellt und diskutiert. Zusätzlich zu den ursachenspezifischen Handlungsempfehlungen werden im Rahmen der Analyse bestehender Sicherheitsnachweise Empfehlungen identifiziert, durch die ein sicheres Systemdesign zu erreichen ist. Diese werden ebenfalls der Liste an Empfehlungen zur Sicherstellung einer ausreichenden Qualität der eingesetzten Daten, Methoden und Prozesse angehängt. Die Vorstellung der Empfehlungen erfolgt nicht ursachenbezogen, sondern orientiert sich am Entwicklungsprozess von gelernten Modellen, um eine Anwendung der Empfehlungen zu erleichtern. Die Ursache oder der Sicherheitshinweis, aus der die Empfehlung resultiert, ist aus Gründen der Nachvollziehbarkeit nach der Beschreibung der Empfehlung angegeben. Im Rahmen der entsprechenden Ursachen in Unterkapitel 4.2 bzw. der angegebenen Abschnitte finden sich weitere Informationen zur Umsetzung der Empfehlung.

Empfehlungen zur Vermeidung der Ursachen fehlender Generalisierung:

- Datenanalyse/ Datenvisualisierung
    - hinsichtlich unterrepräsentierter Klassen. Falls die Unterrepräsentanz nicht vermeidbar/ ungewollt ist, ist diese Eigenschaft entsprechend bestehender Verfahren zu berücksichtigen. (E12)
    - hinsichtlich der Datenverteilung und –dichte, um eine ausreichende Repräsentativität und Menge relevanter Zusammenhänge sicherzustellen (E13, E15)
    - hinsichtlich Ausreißern (E14)

    - hinsichtlich vorhandener Messfehler (E16)
- Aufstellen und Überprüfen von Anforderungen an den Datensatz (E15)
- Unabhängige Überprüfung der Label
    - hinsichtlich ihrer Korrektheit und Genauigkeit (nachträglich generierte Label) (E17)
    - hinsichtlich ihrer Aussagekraft (E17)
- Nutzung von geeigneten Methoden zur Aufteilung der Trainings- und Validierungsdaten (E27)
- Überprüfung der Programmiersprache auf ggf. getroffene Vereinfachungen (E26)
- Nutzung von vorimplementierten Algorithmen (E23)
- Nutzung von vorimplementierten Initialisierungsmethoden (E22)
- Nutzung von Regularisierungsansätzen zur Vermeidung von Overfitting (E22)

Allgemeine Empfehlungen zur Entwicklung eines sicheren Systems:

- Nutzung von ML nur dann, wenn keine vollständige Spezifikation der Aufgabe möglich ist (Abschnitt 3.2.2.3)
- Erweiterung des Datensatzes um synthetische Daten, um die Menge der „bekannten Unbekannten“ zu adressieren, wenn für ML valide Werkzeuge zur Erzeugung dieser Daten zur Verfügung stehen (Abschnitt 3.2.2.3)
- Parallele Entwicklung transparenter Algorithmen, um diese für einen späteren Vergleich mit performanteren Algorithmen hinsichtlich des Aufwand-Nutzen-Verhältnisses zum Führen eines Sicherheitsnachweises zu nutzen (Abschnitt 3.2.1.2, 3.2.2.1, 3.2.1.3, 3.2.2.3)
- Implementierung von Unsicherheits- bzw. Zuverlässigkeitsmetriken der Vorhersage (Abschnitt 3.2.2.1, 3.2.2.3)
- Einsatz von Redundanzen und Fehlerkompensationsmaßnahmen auf Architekturebene (Abschnitt 3.2.2.3)

5.3 Direkte Überprüfungsmethoden

Während bzw. nach der Entwicklung des gelernten Modells sind die folgenden Methoden anzuwenden, um das Vorliegen von Ursachen fehlender Generalisierung, die zuvor durch die Handlungsempfehlungen nicht vermieden wurden, zu untersuchen. Die zu der Überprüfung zugehörigen Ursachen sind hinter den Methoden angegeben. In Unterkapitel 4.2 werden die einzelnen Methoden im Rahmen der jeweiligen Ursache näher erläutert. Wenn das

Vorliegen einer Ursache identifiziert wird, ist diese Ursache, wenn möglich, zu beheben und andernfalls zu analysieren, welche Auswirkungen diese Ursache auf die Generalisierbarkeit besitzt und ob diese für den Betrieb akzeptierbar ist.

- Zur Identifikation von Overfitting/ Underfitting:
    - Beachtung des Klassifizierungsfehlers während der Validierungsphase (gelabelte Daten) bei steigender Algorithmenkomplexität (E22)
    - Beachtung anderer Fehlermetriken während des Trainings (ungelabelte Daten) bei steigender Algorithmenkomplexität (E22)
- Kritisches Betrachten der Ergebnisse des Testdatensatzes,
    - um falsch definierte Funktionen zu identifizieren, (E23)
    - um Fehler in der Berechnung der Funktionen zu identifizieren. (E26)
- Einsatz von Visualisierungsmethoden, um das Verständnis für die Vorgänge im Algorithmus zu erhöhen und unzureichend oder falsch gelernte Zusammenhänge zu identifizieren (E21)
- Variierung der Initialisierung der „objective function“, um das Vorliegen eines lokalen Optimums zu überprüfen (E25)
- Überprüfung auf Datenpunkte, die eine falsche Vorhersage hervorrufen (E14)
- Überprüfung der Sensitivität auf adverserial examples (E24)
- Implementierung des gleichen Algorithmus in zwei unterschiedlichen Implementierungsumgebungen bzw. -sprachen mit anschließendem Vergleich der Ergebnisse, um Vereinfachungen der Implementierungssprache/-umgebung zu identifizieren (E26)

5.4 Funktionale Anforderungen

Um die verbleibenden Ursachen fehlender Generalisierbarkeit zu adressieren, sind die problem- bzw. aufgabenspezifischen funktionalen Anforderungen zu definieren. Für jede Eingangsgröße ist mindestens eine Anforderung in Bezug auf die Ausgangsgröße zu definieren, um ein überprüfbares Soll-Verhalten des gelernten Modells zu generieren. Diese funktionalen Anforderungen stammen aus bisherigen Erfahrungen mit ähnlichen Funktionen oder aus der Fachliteratur. Wird für eine Eingangsgröße nicht mindestens eine funktionale Anforderung aus der Fachliteratur oder sonstigen Erfahrungswerten extrahiert, ist zu begründen, worauf die Sensitivität der Ausgangsgröße auf diese spezielle Eingangsgröße

beruht und diese Begründung dann in Form einer Anforderung zu definieren. Es ist darauf zu achten, dass dieser Sensitivität eine Kausalität und keine Korrelation zugrunde liegt.[245]

Es wird erwartet, dass sich die aufgestellten funktionalen Anforderungen im gelernten Modell wiederfinden, da diese Anforderungen die Realität widerspiegeln und das gelernte Modell auf realen Daten trainiert wird. Finden sich diese Anforderungen nicht wieder, so ist dies ein Hinweis auf das Erlernen falscher Zusammenhänge durch das Modell bzw. das Fehlen relevanter Zusammenhänge im Modell und hierauf basierend einer fehlenden Generalisierbarkeit.

Zur Aufstellung der Anforderungen besteht jedoch die Problematik, dass ML vor allem dort eingesetzt wird, wo eine Spezifikation des Verhaltens nicht vollständig möglich ist.[246] Um dennoch möglichst viele Anforderungen zu erhalten, schlagen Salay und Czarnecki[246] die Verwendung von verschiedenen zusätzlichen Methoden als Ergänzung zu herkömmlicher Anforderungsspezifikation vor:

- Vorher- und Nachher-Bedingungen: Wenn ein Eingangsgrößendatenpunkt eine sog. Vorher-Bedingung erfüllt, dann hat dieser Datenpunkt im gelernten Modell die sog. Nachher-Bedingung in Form einer bestimmten Ausgangsgröße zu erfüllen. Ein Beispiel im Rahmen einer bildbasierten Fußgängerdetektion ist, dass jedes Objekt, was zwei Beine, zwei Arme, einen Torso und einen Kopf als Gesamtverband besitzt und dabei aufrecht steht (Vorher-Bedingung) ein Fußgänger darstellt (Nachher-Bedingung). Dabei gilt die Bedingung jedoch nur in diese Richtung, denn nicht jeder Fußgänger steht beispielsweise aufrecht. Durch diese Art der Bedingungen ist es jedoch möglich, das Verhalten teilweise zu überprüfen, auch wenn nicht alle Vorher-Bedingungen bekannt sind, die zu einer Nachher-Bedingung führen müssen.
- Äquivarianz und Invarianz: Im Rahmen einer invarianten Anforderung wird definiert, welche Möglichkeiten bestehen, dass die Eingangsgröße sich ändert, ohne dass die zugehörige Ausgangsgröße verändert wird. Ein Beispiel hierfür stellt ein Verkehrsschild dar, welches seine Bedeutung (z.B. Überholverbot) nicht ändert, egal, ob es mit Schnee bedeckt ist, oder nicht. Äquivariante Anforderungen zeichnen sich dadurch aus, dass sie definieren, dass die Änderung einer Eingangsgröße eine bestimmte Änderung der zugehörigen Ausgangsgröße hervorruft.
- Andere Arten der Spezifikation:
    - Musterbasierte Bedingungen: Z. B. eine Kontur eines Fußgängers, in die ein Fußgängerbild zu passen hat. Dieses Muster ist jedoch formal zu definieren.
    - Kontextbasierte Bedingungen: Ein Fußgänger hat sich beispielsweise innerhalb von X Metern in der Nähe der Straße und sich nicht in einem Schau-

[245] Siehe Abschnitt 3.2.2.3.

[246] Vgl. Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018), S. 7f.

fenster usw. zu befinden. Im Allgemeinen erfordern diese Bedingungen, dass die Ein- und Ausgangsgrößen in eine breitere situative Darstellung, die Kontextinformationen enthält, eingebettet sind.

Die Überprüfung der funktionalen Anforderungen erfolgt entweder durch Testfälle oder der Darstellung der geforderten Zusammenhänge zwischen den Eingangsgrößen und Ausgangsgröße(n).

Wie bereits diskutiert, reicht jedoch die Überprüfung der funktionalen Anforderungen alleine nicht aus, um eine ausreichende Generalisierbarkeit zu bestätigen. Neben der bereits in Unterkapitel 5.1 diskutierten Problematik der möglichen Unvollständigkeit der funktionalen Anforderungen besitzen die bekannten, für den Menschen interpretierbaren funktionalen Anforderungen wenige Dimensionen, wodurch das gelernte Modell zur Überprüfung dieser Anforderungen nur in diesen Dimensionen betrachtet wird. Durch dieses Vorgehen wird die Höherdimensionalität des Modells missachtet, wodurch es möglich ist, dass trotz Erfüllen aller funktionalen Anforderungen ein Fehlverhalten des Modells auftritt. Zusätzlich wird die Kombination der Eingangsgrößen, für die keine funktionalen Anforderungen vorliegen, nicht auf ihre Sinnhaftigkeit hin überprüft. Hierdurch ist es notwendig die funktionalen Anforderungen durch die in Unterkapitel 5.5 vorgestellten Robustheitsanforderungen zu ergänzen.

5.5 Robustheitsanforderungen

Im Gegensatz zu funktionalen Anforderungen sind Robustheitsanforderungen funktionsunspezifisch und dadurch allgemeingültig definierbar. Die Robustheitsanforderungen dienen der Überprüfung impliziter Anforderungen, die mit der Eigenschaft Generalisierbarkeit einhergehen, wie beispielsweise, dass das funktionale Verhalten eines gelernten Modells nicht an einzelnen wenigen Datenpunkten des gesamten Trainingsdatensatz hängt. Dass dieses Beispiel dem Begriff „Robustheit“ zugeordnet wird, liegt daran, dass eine Änderung des Trainingsdatensatzes um wenige Punkte, keine Änderung des funktionalen Verhaltens hervorzurufen hat – dementsprechend „robust“ gegenüber Änderungen im Entwicklungsprozess bzw. den verwendeten Daten ist. Im Rahmen von konventionell programmierten Modellen werden diese Robustheits- bzw. impliziten Anforderungen der Generalisierbarkeit stets erfüllt, da die für den Betriebsbereich gültigen Zusammenhänge explizit programmiert werden. Wie bereits erwähnt, werden diese impliziten Anforderungen der Generalisierbarkeit mit der Frage „Wie ist es möglich, das gelernte Modell durch Testfälle oder Änderungen im Entwicklungsprozess so zu stören, so dass falsche Vorhersagen getroffen werden bzw. die benötigte Generalisierbarkeit nicht mehr vorliegt?“ abgeleitet. Die bisher aus Literatur und der vorangegangenen Analyse der Ursachen fehlender Generalisierbarkeit identifizierten Robustheitsanforderungen werden im Folgenden vorgestellt. Diese Auflistung enthält keinen Anspruch auf Vollständigkeit und ist bei Bedarf zu erweitern.

Datenquantität: Die Quantität der Daten besitzt einen Einfluss auf die Generalisierbarkeit des Modells.[247] Um festzustellen, wie sensitiv das gelernte Modell auf die Veränderung der Quantität reagiert, werden zwei Anforderungen mit unterschiedlichem Fokus untersucht:

- DQ1: Die gleichmäßige Änderung (bspw. über alle Klassen hinweg) der Datensatzgröße in den unterschiedlichen Entwicklungsphasen bis zu einem gewissen Schwellwert besitzt keine Auswirkungen auf die grundlegende Funktionalität des Modells.
- DQ2: Alle beabsichtigten Klassen sind innerhalb des Trainingsdatensatzes für die grundlegende Funktionalität des Modells hinreichend vertreten. Die Veränderung der Klassenrepräsentanz einzelner Klassen verändert die Leistungsfähigkeit für jede andere beabsichtigte Klasse nicht.

Datenvorverarbeitung: Durch die Änderung der Vorverarbeitung der Trainingsdaten, wie beispielsweise das Einbringen eines künstlichen Rauschens, ist es möglich, dass bei fehlender Generalisierbarkeit ein Modell mit funktional falschem Verhalten resultiert.[248] Durch die künstliche Erzeugung dieser Änderung sind daher Erkenntnisse über die Robustheit des Modells bzw. die Generalisierbarkeit des Modells möglich. Diese Änderung der Vorverarbeitung hat dabei lediglich mikroskopische Auswirkungen im Vergleich Trainingsdatensatz des finalen Modells zu enthalten, da durch eine zu starke Änderung der Eingangsdaten die inhärenten Zusammenhänge verändert werden. Durch das Prinzip der Induktion des Maschinellen Lernens bedingt resultiert sonst durch die Änderung dieser inhärenten Zusammenhänge zwangsweise eine Änderung der Funktionalität des Modells, die den Rahmen der beabsichtigten Robustheitsüberprüfung übersteigt. Die Anforderung lautet wie folgt:

- DV1: Mikroskopische Veränderungen der Vorverarbeitung der Eingangsdaten des Modells besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells.

Abdeckung: Durch die begrenzte, endliche Anzahl an Daten, die dem ML zur Verfügung gestellt wird, und dem viel größeren Raum, den mögliche Betriebsdaten einnehmen, existieren Bereiche, die eine geringere Abdeckungsrate besitzen. Dies sind beispielsweise Randbereiche, die auch im späteren Betrieb selten erreicht werden und daher auch in den Trainingsdaten eine spärliche Abdeckung besitzen. Hierdurch sind wenige oder keine Beispiele zur Generierung induktiver Folgerungen verfügbar, wodurch die Funktionalität in diesen Bereichen aus der Extrapolation der dichter belegten Bereiche der Trainingsdaten resultiert. Deshalb sind diese Bereiche im Betrieb besonders anfällig für Fehlverhalten,

[247] Siehe E12: Quantität einer Klasse zu gering und E13: Quantität der relevanten Zusammenhänge zu gering in Abschnitt 4.1.1

[248] Siehe E16: Messfehler verdeckt relevante Zusammenhänge zu stark in Abschnitt 4.1.1

weshalb es deren Funktionalität gesondert zu untersuchen gilt.[249] Die zugehörige Anforderung lautet:

- A1: Die korrekte Funktion des Modells ist auch in dessen Bereichen mit einer spärlichen Abdeckungsrate durch die Trainingsdaten gewährleistet.

Trainingsprozess: Der Trainingsprozess beginnt mit dem Initialisierungsprozess, welcher anschließend in die Zuführung der Trainingsdaten in den Algorithmus mündet. Aus diesen beiden Punkten wird je eine Robustheitsanforderung extrahiert.

Die Wahl der Initialisierungsparameter bestimmt einerseits den Rechenaufwand, der notwendig ist, um zu einem Optimum der „objective function“ zu gelangen, andererseits wird hierdurch ebenfalls bei nicht-konvexen Kostenfunktionen bestimmt, ob ein globales Optimum erreicht wird oder lediglich ein lokales.[250] Werden mit zwei unterschiedlichen Initialisierungen zwei unterschiedliche Modelle unter sonst identischen Bedingungen trainiert, resultiert mindestens eines der beiden Modelle aus einem lokalen Optimum. Bereits im Rahmen der Überprüfung der direkten Anforderungen wird das Vorliegen eines lokalen Optimums durch Variation der Initialisierungsparameter bzw. des Initialisierungsprozesses untersucht.[251] Von Initialisierungsprozess ist die Sprache, da bereits vorgefertigte Implementierungen im Rahmen einiger Algorithmen existieren, die eine *n*-fache zufällige Initialisierung vornehmen und das Training von *n* Modellen parallel durchgeführt wird. Aus dieser Modellschar wird anschließend das Optimum ausgewählt.[252] Bei einer ausreichend großen Menge *n* ist daher das globale Optimum stets enthalten. Fällt diese Überprüfung negativ aus, ist die folgende Anforderung T1 bereits erfüllt. Jedoch wird diese Überprüfung beispielsweise bei Neuronalen Netzen sehr häufig positiv ausfallen, da diese zu lokalen Optima tendieren.[253] Die *n*-fache zufällige Initialisierung ist in diesem Fall durch den hohen Rechenaufwand nicht realisierbar. Hierdurch sind weitere Analysen hinsichtlich der Robustheit des gelernten Modells auf unterschiedliche Initialisierungsparameter durchzuführen, um festzustellen, ob zumindest Modelle mit der grundlegend gleichen Funktionalität aus unterschiedlichen Initialisierungen resultieren, wenn sie nicht dieselbe Struktur besitzen. Die zugehörige Anforderung T1 lautet:

- T1: Der Initialisierungsprozess der Algorithmen führt zu Modellen, die die grundlegend gleiche Funktionalität besitzen.

Maschinelle Lernalgorithmen unterscheiden sich unter anderem auch in der Art der Zuführung ihrer Trainingsdaten. Entweder werden dem Algorithmus alle Daten gleichzeitig zugeführt, wie im Fall eines K-Menas-Clusterings[252], oder der Algorithmus optimiert seine

[249] Siehe Abschnitt 3.2.2.3, „unbekannte Unbekannte“.

[250] Siehe E25: Lokales statt globalen Minimum erreicht in Abschnitt 4.1.2.

[251] Siehe Unterkapitel 5.3.

[252] Vgl. scikit-learn: sklearn.cluster.KMeans (2019).

[253] Vgl. Akarachai, A.; Daricha, S.: Avoiding Local Minima (2007), S. 100.

Parameter nach einzelnen Untermengen des Gesamtdatensatzes, wobei die Anzahl der Datenpunkte in einer Untermenge wählbar ist. Ein Beispiel hierfür stellen Neuronale Netze dar.[254] Durch eine Veränderung der Reihenfolge in der Zuführung dieser Untermengen wird eine Variation des Trainingsprozesses erreicht. Beim Vorliegen einer hohen Generalisierbarkeit des gelernten Modells wird auch bei Veränderung dieser Reihenfolge die gleiche grundlegende Funktionalität erwartet, woraus sich die folgende Anforderung ableitet:

- T2: Veränderungen der Datensequenzen während des Trainingsprozesses besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells.

Die Überprüfung der vorgestellten Robustheitsanforderungen beruht teilweise auf Testfällen (d.h. neue, „ungesehene" Eingangsdaten) oder auch auf niedrigdimensionaler Visualisierungen des Datensatzes oder Modells. Darüber hinaus besteht die Möglichkeit den Trainingsdatensatz zu verändern und anschließend das Modell erneut zu trainieren. Ein Vergleich zwischen neu-trainiertem und ursprünglichem Modell gibt anschließend Aufschlüsse über die Robustheit bzw. die Generalisierbarkeit. Dieser Vergleich erfolgt beispielsweise durch die erreichte Leistungsfähigkeit in der Vorhersage der Ausgangsgrößen, wenn diese im Fall eines Supervised-Learning-Ansatzes vorliegen. Im Bereich des Unsupervised-Learning ist der relative Vergleich zwischen zwei Modellvorhersagen möglich.

Die Robustheitsanforderungen werden durch die gelernten Modelle häufig nicht vollständig erfüllt, was allerdings auch keine zwangsweise Zielvorgabe darstellt. Daher stellen sie keine Festforderung, die es notwendigerweise zu erfüllen gilt, sondern eine Zielforderung, auch Optimalitätsforderung genannt, dar.[255] Ein Beispiel hierfür besteht in der Änderung der Vorverarbeitung der Daten. Bis zu einem gewissen Grad der Änderung der Vorverarbeitung (beispielsweise eine veränderte Abstimmung der Filterung) wird sich die grundlegende Funktionalität des gelernten Modells nicht ändern. Ab einer zu starken Änderung der Vorverarbeitung wird diese Änderung jedoch unweigerlich einen Einfluss auf die Funktionalität besitzen. Wichtig ist im Rahmen der Überprüfung der Anforderungen daher nicht, dass die Robustheitsanforderungen vollständig erfüllt werden, sondern welche Schlüsse sich aufgrund der Überprüfung auf die Generalisierbarkeit des Modells ziehen lassen bzw. welchen Grad der Erfüllung dieser Anforderungen erreicht werden. Hierdurch begründet enthalten die oben vorgestellten Robustheitsanforderungen keine quantitativen Angaben, wie beispielsweise bis zu welchem Schwellwert eine Anforderung als erfüllt angesehen wird, sondern sind bewusst offen formuliert.

---

[254] Vgl. Brownlee, J.: What is the Difference Between a Batch and an Epoch in a Neural Network? (2018).

[255] Vgl. Mattmann, I.: Dissertation, Modellintegrierte Produkt- und Prozessentwicklung (2017), S. 60.

6 Prototypische Anwendung

Nachdem in Kapitel 5 ein strukturierter Ansatz zur Überprüfung der Generalisierbarkeit bzw. zur Identifikation fehlender Generalisierbarkeit abgeleitet wurde, ist dessen Anwendbarkeit zu untersuchen. Hierdurch wird der Forschungsfrage *„Ist der Ansatz praktisch anwendbar?“* aus Unterkapitel 1.2 nachgegangen. Zur Beantwortung wird die Hypothese *„Der Ansatz ist ohne Einschränkungen anwendbar“* aufgestellt. Da bisher keine Erfahrungen mit der Anwendung des Ansatzes bestehen, ist es nicht möglich, einen Anwendungsfall auszuwählen, der diese Hypothese besonders herausfordert. Deshalb wird im Rahmen dieser Arbeit ein singularer Anwendungsfall gewählt, durch welchen erste Erkenntnisse über die Anwendung des Ansatzes gewonnen werden. Wird die Hypothese der Anwendbarkeit durch diesen beliebigen Anwendungsfall nicht falsifiziert, stellt diese erste Anwendung und die gewonnenen Erkenntnisse einen Ausgangspunkt für weitere Falsifikationstests dar.

Der ausgewählte, sicherheitsrelevante Anwendungsfall aus dem Bereich der Fahrerassistenzsysteme wird in Unterkapitel 6.1 vorgestellt. Dieser Anwendungsfall verwendet ein gelerntes Modell zur Fahrstilerkennung, auf welches das Verfahren zur Überprüfung der Generalisierbarkeit in Unterkapitel 6.2 angewendet wird. Im Fazit (Unterkapitel 6.3) wird anschließend die Forschungsfrage zusammengefasst beantwortet.

6.1 Übersicht Anwendungsfall

Im Forschungsprojekt „PRORETA 4“[256], welches unter dem Motto „Safety by Learning“ steht, wurde ein Fahrerassistenzsystem namens Stadtassistent bzw. City Safety Assistant entwickelt, das Empfehlungen und Warnungen für Manöverausführungen in typischen städtischen Situationen ausgibt. Diese Empfehlungen werden mithilfe eines gelernten Modells zur Fahrstilidentifikation an den aktuellen Fahrer angepasst. Der Stadtassistent wurde in drei Use-Cases implementiert: Linksabbiegen bei entgegenkommenden vorfahrtsberechtigtem Verkehr, die Einfahrt in einen Kreisverkehr und die Annäherung bzw. die Überfahrt einer Kreuzung mit der Vorfahrtsregelung rechts-vor-links. Im Folgenden wird sich auf den Use-Case Linksabbiegen konzentriert. Nähert sich der Fahrer einer Kreuzung, an der ein Linksabbiegen mit Vorfahrt des Gegenverkehrs möglich ist, und blinkt der Fahrer, wird das System aktiviert. Es visualisiert die Lücken im Gegenverkehr in einem Display, welches sich anstelle herkömmlicher Instrumenten-Cluster hinter dem Lenkrad befindet (siehe Abbildung 6-1).

[256] www.poreta.de.

Abbildung 6-1: Stadtassistent aktiv im Use-Case Linksabbiegen

Wird die aktuell vorhandene Lücke im Gegenverkehr für den individuellen Fahrer zum Linksabbiegen empfohlen, wird diese grün eingefärbt, andernfalls rot. Ein zusätzlicher Pfeil am Beginn der aktuellen Lücke verdeutlicht diese Empfehlung. Der Stadtassistent wurde in einen Versuchsträger implementiert und die Funktionalität im öffentlichen Straßenverkehr auf einer Demonstrationsveranstaltung vorgeführt.

Im Folgenden wird zunächst auf die generelle Funktionsweise des Gesamtsystems eingegangen, um anschließend die Eignung als Anwendungsfall durch Ableitung der Sicherheitsrelevanz des gelernten Modells der Fahrstildetektion in Abschnitt 6.1.2 zu zeigen. Wie die zum Training genutzten Daten erhoben wurden, wird in Abschnitt 6.1.3 beschrieben, um hierauf basierend Details zum gelernten Modell in Abschnitt 6.1.4 vorzustellen.

6.1.1 Funktionsweise

Die prinzipielle Funktionsweise des Stadtassistenten in den Use-Cases Linksabbiegen und Einfahrt in den Kreisverkehr ist mit Abbildung 6-2 gegeben. Während der Fahrt wird fortlaufend analysiert, ob aktuell eines der drei Manöver vorliegt, die zur Fahrstilidentifikation genutzt werden. Diese drei Manöver sind die Annäherung an ein Linksabbiegen, der Abbiegevorgang an sich und die Fahrt durch einen Kreisverkehr. In diesen Manövern erfolgt die Zuordnung des Fahrers zu einem Cluster, der jeweils einen Fahrstil repräsentiert. Aus den Zugehörigkeitswerten des Fahrers zu den drei Clustern aus den Manövern wird ein Gesamtfahrverhalten gebildet. Aus diesem leitet sich der Schwellwert ab, ab dem die Empfehlung für die Ausführung eines Manövers ausgesprochen wird. Dieser Schwellwert stellt im Fall des Linksabbiege-Use-Cases die Lücke im Gegenverkehr dar, ab dem ein Abbiegen für den aktuell gezeigten Fahrstil sicher möglich ist. Mit dem Abgleich der aktuellen Situation wird, wie bereits beschrieben, eine Empfehlung für oder gegen die Wahl der Lücke zum Abbiegen gegeben.

Abbildung 6-2: Funktionsaufbau Stadtassistent

Damit der Schwellwert, ab dem die Lückengröße empfohlen wird, nicht von der Geschwindigkeit der beteiligten Fahrzeuge abhängt, werden die Lücken im Gegenverkehr und die Empfehlungsschwellwerte zeitbasiert angegeben. Für die erste Lücke wird hierzu der Quotient aus dem Abstand, den das erste Fahrzeug des Gegenverkehrs bis zur Mitte der Zielkreuzung besitzt, durch dessen aktuelle Fahrgeschwindigkeit geteilt (siehe Abbildung 6-3). Es besteht hierbei die Annahme, dass die aktuelle Geschwindigkeit des entgegenkommenden Fahrzeugs konstant bleibt. Die maximal mögliche Lücke wird durch die Sensorreichweite des Radars, welcher zur Bestimmung der Position und der Geschwindigkeit des Gegenverkehrs genutzt wird, und die in diesem Bereich erlaubte Geschwindigkeit plus 10% bestimmt. Die minimale Zeitlücke, bei der ein Fahrer aus dem Probandenpool an einer bestimmten Kreuzung den Abbiegevorgang eingeleitet hat, liegt bei ca. 3,5 Sekunden. Auch sportliche Fahrer benötigen ca. ein bis zwei Sekunden, um in die Kreuzung nach einem Stillstand zu überqueren. Dabei wird die Zeitlücke, die ein Fahrer „akzeptiert“, d.h. in der er einbiegt, zum Zeitpunkt des Losfahrens aus dem Stillstand gemessen.

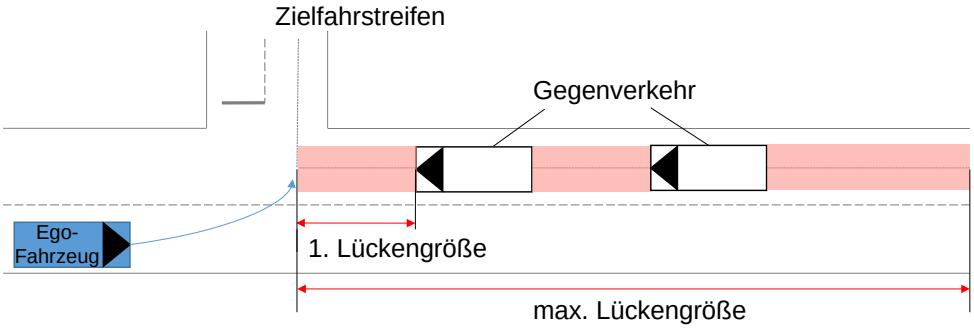


Abbildung 6-3: Schematische Darstellung Use-Case Linksabbiegen

6.1.2 Sicherheitsrelevanz

Damit sich das gelernte Modell, die Fahrstildetektion, als Anwendungsfall eignet, hat dieses eine Sicherheitsrelevanz zu besitzen, damit die Notwendigkeit der Erfüllung eines Sicherheitsnachweises besteht.[257] Zur Feststellung dieser Sicherheitsrelevanz wurde entsprechend dem Vorgehen der ISO 26262[258] eine Gefahrenanalyse des Systems durchgeführt (siehe Anhang 8-3).[259] Die betrachteten Systemfunktionen sind dabei „Empfehlung „rot“ ausgeben“, „Empfehlung „grün“ ausgeben“ und „Keine Empfehlung ausgeben, da Lernmodus (fehlende Konfidenz)“. Als mögliche Fehlfunktionen wurden eine zu sportliche

[257] Siehe Unterkapitel 2.1.

[258] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

[259] Gefahren, die durch eine System-Nutzer-Interaktion entstehen und durch die ISO/ PAS 21448 adressiert werden, werden nicht betrachtet. Das Ziel der Sicherheitsanalyse besteht lediglich darin, die generelle Sicherheitsrelevanz des gelernten Modells zu identifizieren, um die Eignung als Anwendungsfall herauszustellen, und keinen Sicherheitsnachweis des Gesamtsystems zu führen.

oder eine zu konservative Empfehlung der Zeitlücken im Gegenverkehr identifiziert. Darüber hinaus besteht ebenfalls die Möglichkeit der Fehlfunktion, dass die Konstanz der Empfehlung zu gering ist, d.h. dass die Empfehlung zu stark bzw. schnell zwischen „rot“ und „grün“ wechselt. Es wurden insgesamt fünf Betriebszustände in Kombination mit zwei Fahrsituationen betrachtet, woraus 27 unterschiedliche Kombinationsmöglichkeiten resultieren. Es ergeben sich zehn mögliche Gefährdungen, die in einer Fehlermöglichkeits- und –einflussanalyse (FMEA) hinsichtlich ihres Risikos bewertet wurden, um hierauf basierend die für dieses Sicherheitsrisiko passenden Methoden in der Entwicklung auszuwählen. Die Fehlfunktion „Konstanz der Empfehlung zu gering“ sowie „zu sportliche Eingruppierung des Fahrers“ besitzen die höchste Risikoeingruppierung des Systems (ASIL B). Basierend auf dieser Einstufung werden Sicherheitsziele abgeleitet sowie ein Sicherheitskonzept entwickelt (siehe Anhang A.3). Die Wahrscheinlichkeit sowie die potentiellen Folgen der Fehlfunktion „Konstanz der Empfehlung zu gering“ werden durch die Implementierung einer zusätzlichen Pufferzeit auf den individuellen Schwellwert zur Empfehlung der Linksabbiege-Zeitlücke vermindert.

Allerdings wird innerhalb dieses Sicherheitskonzepts festgestellt, dass, falls dem Fahrer ein zu kleiner Lückenschwellwert zugeordnet wird, ein Sicherheitsrisiko für den Fahrer entsteht, welches nicht durch Vermeidungsmaßnahmen zu beheben bzw. die Wahrscheinlichkeit des Auftretens zu vermindern ist. Die Ursache des zu geringen Lückenschwellwerts liegt in der inkorrekten Identifikation seines Fahrstils. Zwar handelt es sich beim Stadtassistenten lediglich um ein Empfehlungssystem und die Verantwortung der Fahrzeugführung verbleibt beim Fahrer, jedoch wird erwartet, dass sich der Fahrer nach einiger Nutzungszeit auf das System verlässt. Da die Zuordnung des Fahrers entsprechend des gezeigten Fahrstils im gelernten Modell stattfindet, ist die Sicherheit dieses Modells hinsichtlich der nicht zu sportlichen Einordnung des Fahrers zu beweisen, woraus sich die Eignung dieses gelernten Modells als sicherheitsrelevanter Anwendungsfall ableitet.

6.1.3 Trainingsdaten

Zum Training der Fahrstildetektion wird ein Datensatz verwendet, welcher speziell für diese Systementwicklung erhoben wurde. Es absolvierten 32 unterschiedliche Fahrer die gleiche Strecke 30-mal, um sowohl eine hinreichende Menge an Daten für deren Fahrstil als auch genügend Informationen über die Lückenakzeptanz beim Linksabbiegen zu erhalten. Die Strecke (siehe Anhang 8-1) befindet sich in einem Wohn-Gewerbe-Mischgebiet und wurde systematisch für diesen Anwendungsfall durch Rausch[260] ausgewählt. Sie enthält drei Linksabbiege-Vorgänge, bei denen dem entgegenkommenden Verkehr Vorfahrt zu gewähren ist, sowie eine Kreisverkehrsdurchfahrt. Für eine Streckendurchfahrt wurden 3 – 4 Minuten benötigt. Um eine hohe Repräsentativität der Probanden hinsichtlich verschiedener Fahrstile zu erhalten, wurden die Teilnehmer entsprechend der Kategorien Al-

[260] Rausch, S.: Master-Thesis, Ableitung einer Simulationsumgebung aus der realen Welt (2016).

ter, Geschlecht und Fahrerfahrung ausgewählt. Die Kategorie Fahrerfahrung als nicht direkt messbare Größe wird aus den Indikatoren Dauer des Führerscheinbesitzes, jährliche Fahrleistung sowie Gesamtfahrleistung abgeschätzt. Die Verteilung der Eigenschaften ist Anhang 8-2 zu entnehmen. Die Prozesskette von den Rohdaten zu den einzelnen Trainingsdatenpunkten für das gelernte Modell der Fahrstildetektion ist in Anhang A.4 beschrieben.

6.1.4 Fahrstilzuordnung

Die Fahrstilzuordnung je Manöver basiert auf einem Clustering, welcher die Zugehörigkeit des aktuellen Fahrers zu einem Cluster basierend auf dem gezeigten Fahrstil ausgibt (Variante A, Abbildung 6-4). Es besteht prinzipiell die Möglichkeit, neben der Zugehörigkeit des gezeigten Fahrstils zu dem wahrscheinlichsten Cluster die Zugehörigkeitswerte zu jedem einzelnen Cluster anzugeben, um hieraus den Gesamtfahrstil zu berechnen (Variante B, Abbildung 6-4).[261] Allerdings wurde sich aufgrund der schnelleren Adaptivität auf Änderungen des Fahrstils für die Variante A in der finalen Systemauslegung entschieden.

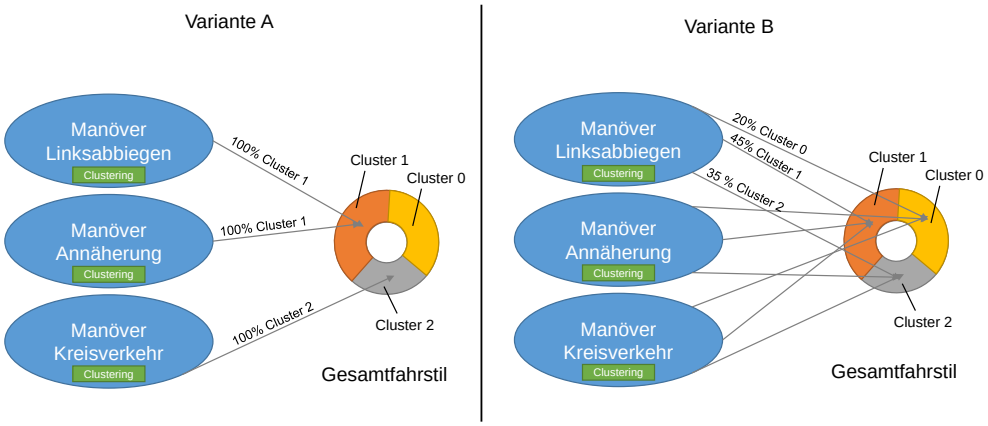


Abbildung 6-4: Varianten der Zusammensetzung des Gesamtfahrstils

Manöver, die eine geringere zeitliche Distanz zum aktuellen Zeitpunkt besitzen, gehen dabei stärker in den Gesamtfahrstil ein. Weiter in der Vergangenheit liegende Manöver werden ab einem gewissen Punkt nicht mehr im Fahrstil berücksichtigt, wodurch eine Art des „Vergessens“ implementiert ist, damit sich die aus dem Gesamtfahrstil resultierende Manöverempfehlung zum Abbiegen stets auf den aktuell gezeigten Fahrstil des Fahrers bezieht.

Ein Clustering-Ansatz wurde gewählt, da die Label der einzelnen Fahrer des Trainingsdatensatzes zwar durch eine Selbstevaluation der Fahrer vorhanden sind, aber aufgrund

[261] Die Wahrscheinlichkeit einer Clusterzugehörigkeit wird durch die Distanz des Datenpunkts zu den Clusterschwerpunkten berechnet. Auf den genauen Clusteringalgorithmus wird in Abschnitt 6.2.1 eingegangen.

ihrer hohen Subjektivität als nicht vertrauenswürdig eingestuft wurden.[262] Hierdurch ist es notwendig, auf eine andere Weise den Zusammenhang zwischen Cluster und Fahrstil herzustellen, da ohne diesen Zusammenhang zwar ein Gesamtfahrstil basierend auf den drei Clustern abgeleitet werden kann, dieser jedoch keine Aussagekraft für eine Schwellwertfindung besitzt. Wie dieser Zusammenhang hergestellt wird, ist in Abschnitt 6.2.1 erläutert.

6.2 Überprüfung

Die Anwendung des in Unterkapitel 5.1 vorgestellten Ansatzes zur Überprüfung der Generalisierbarkeit fokussiert sich lediglich auf einen Teil des gelernten Modells zur Fahrstilerkennung. Durch die vorherrschende Trennung des Fahrstilmodells in drei einzelne gelernte Manövermodelle ist es möglich, jedes der gelernten Teilmodelle bezüglich ihrer Sicherheit unabhängig von den anderen zu betrachten.[263] Da alle drei Teilmodelle auf dem gleichen Grundalgorithmus basieren, birgt zudem die Betrachtung aller Teilmodelle keinen wissenschaftlichen Mehrwert im Vergleich zur Betrachtung eines der Teilmodelle hinsichtlich der Anwendbarkeit des Ansatzes zur Detektion fehlender Generalisierbarkeit. Es wird sich daher im Folgenden auf das Manövermodell „Linksabbiegen“ fokussiert, da die notwendigen Grundlagen dieses Manövers durch die Fokussierung auf den Use-Case Linksabbiegen bereits gelegt wurden.

Die Anwendung des in Unterkapitel 5.1 vorgestellten Ansatzes beschränkt sich auf den dritten (Überprüfung der funktionalen Anforderungen) und vierten Schritt (Überprüfung der Robustheitsanforderungen) des Vorgehens. Beide Schritte sind für die erfolgreiche Anwendung des Ansatzes und demnach zur Beantwortung der Frage *„Ist der Ansatz praktisch anwendbar?“* notwendig (siehe Unterkapitel 1.2), wohingegen die Anwendung des ersten (Sicherstellung der Qualität der Daten, Methoden und Prozesse) und zweiten Schrittes (Direkte Überprüfung von Ursachen) des Ansatzes optional sind. Das liegt darin begründet, dass die ersten beiden Schritte lediglich die Ursachen fehlender Generalisierbarkeit partiell vermeiden bzw. deren Vorliegen identifizieren und hiermit keine Vollständigkeit zu erreichen ist. Durch die Überprüfung der Auswirkungen der fehlenden Generalisierbarkeit im dritten und vierten Schritt werden Ursachen, die in den ersten zwei Schritten nicht adressiert oder „übersehen“ wurden, zusätzlich berücksichtigt. Hierdurch stellen die letzten beiden Schritte die „Showstopper“ der Anwendbarkeit des Ansatzes dar, wodurch sich der Fokus auf diese in den folgenden Abschnitten ergibt. Bevor auf die Anwendung des dritten und vierten Schrittes eingegangen wird, wird in Abschnitt 6.2.1 zunächst das zu überprüfende Manövermodell des Linksabbiegens detaillierter vorgestellt.

---

[262] Hierdurch wurde der Empfehlung zur Überprüfung der Label hinsichtlich ihrer Aussagekraft Rechnung getragen (siehe Unterkapitel 5.2).

[263] Siehe Abschnitt 6.1.2.

6.2.1 Überblick Manövermodell Linksabbiegen

Zum Training der Fahrstildetektion aus Linksabbiege-Vorgängen sind zunächst die einzelnen Abbiegevorgänge aus den zur Verfügung stehenden Daten der Probandenstudie zu extrahieren. Das Manövermodell betrachtet lediglich Linksabbiegevorgänge aus dem Stillstand mit Vorfahrt des Gegenverkehrs. Das finale Manövermodell basiert auf den Größen Geschwindigkeit, Längsbeschleunigung, Querbeschleunigung sowie Lenkradwinkelgeschwindigkeit des Datensatzes. Da der genutzte Clustering-Algorithmus nicht zur Analyse von Zeitreihen geeignet ist, werden aus den Zeitreihen der aufgeführten Signale statistische Merkmale abgeleitet.[264] Da die aufgenommenen Signale des Manövers unterschiedliche Einheiten und Größenordnungen besitzen, werden die verwendeten Merkmale zum Training des Modells standardisiert.[264] Folgende sieben direkt gemessene Größen werden im finalen Stand des Modells, welcher in den nachfolgenden Abschnitten das Überprüfungsobjekt darstellt, genutzt:

- Maximum der Geschwindigkeit
- Maximum der longitudinalen Beschleunigung
- Maximum der lateralen Beschleunigung
- Minimum der lateralen Beschleunigung
- Maximum der Lenkradwinkelgeschwindigkeit
- Standardabweichung der Lenkradwinkelgeschwindigkeit
- Minimum der Lenkradwinkelgeschwindigkeit

Zusätzlich wird die Eingangsgröße Ruck als weiterverarbeitetes Merkmal der longitudinalen Beschleunigung genutzt, da es in der Literatur als leistungsfähiger Indikator des Fahrstils identifiziert wurde.[265] Ruck beschreibt die Änderungsrate einer Beschleunigung und wird daher durch die Ableitung der Beschleunigung berechnet.[266] Folgende statistische Größen werden im finalen Modell als standardisierte Eingangsmerkmale verwendet:

- Maximum des Rucks
- Minimum des Rucks

Der maximale Ruck zeigt an, wie schnell und fest das Gaspedal betätigt wird, um die Kreuzung zu erreichen. Der minimale Ruck (= maximaler negativer Ruck) gibt Hinweise darauf, wie schnell das Gaspedal losgelassen wird. Aktives Bremsen ist bei diesem Manöver nicht Bestandteil eines normalen Fahrverhaltens, da, wie bereits erwähnt, nur das Abbiegen aus dem Stillstand im Manövermodell betrachtet wird. Die Eingangsmerkmale des

[264] Siehe Abschnitt 2.2.1.

[265] Vgl. Murphey, Y. L. et al.: Driver's style classification using jerk analysis (2009).

[266] Vgl. Hamalainen, W. et al.: Jerk-based feature extraction (2011), S. 834.

Rucks wurden erst zu einem späteren Entwicklungsstand des Modells genutzt, um dessen Leistungsfähigkeit zu verbessern.[267]

Zum Clustering wird ein K-Means Algorithmus in der scikit-learn-Implementierung genutzt.[268] Der Algorithmus arbeitet iterativ, um jeden der *n* Datenpunkte basierend auf den Eingangsmerkmalen einer von *k* (im vorliegen Algorithmus drei) Cluster *C* zuzuweisen. Datenpunkte werden durch den Algorithmus basierend auf der Ähnlichkeit der Eingangsmerkmale x_i gruppiert.[269] Die Ähnlichkeit wird durch die Berechnung des Abstands zwischen den Datenpunkten und den sogenannten Clusterschwerpunkten, die durch den Mittelwert der Datenpunkte $\bar{x}_j$, die zu einem Cluster gehören, berechnet werden, beschrieben. Der Algorithmus zielt dabei darauf ab, die Abstände zwischen den Datenpunkten und Clusterschwerpunkten zu minimieren:[270]

$$\sum_{i=0}^{n} \min_{\bar{x}_j \in C} \left(\| x_i - \bar{x}_j \|^2 \right) \tag{6.1}$$

Nach der Initialisierung der Clusterschwerpunkte bestehen die Berechnungen des K-Means-Algorithmus aus einer Schleife von zwei Schritten. Im ersten Schritt wird jeder Datenpunkt dem nächstgelegenen Clusterschwerpunkt zugeordnet. Im zweiten Schritt werden neue Schwerpunkte erstellt, indem die Mittelwerte aller einem vorherigen Clusterschwerpunkt zugewiesenen Datenpunkten berechnet werden. Anschließend wird die Differenz zwischen dem alten und neuen Clusterschwerpunkt berechnet. Der Algorithmus wiederholt diese beiden Schritte, bis diese Differenz unter einem Schwellwert liegt, d.h. bis sich die Clusterschwerpunkte nicht mehr wesentlich bewegen.[270]

Die Ergebnisse des K-Means-Clustering-Algorithmus sind:

- Die Lage der Schwerpunkte der Cluster C_j, wodurch die Zuordnung von neuen Eingangsdaten möglich ist
- Zuordnung der Trainingsdaten[269]

Hierdurch ist jeder Datenpunkt einem einzelnen Cluster zu einem gewissen Wert basierend auf seinem Abstand zum Clustermittelpunkt zugeordnet. Im finalen Modell wird jedoch nur der nächstgelegene Cluster zur Vorhersage des Gesamtfahrstils genutzt, weshalb diese Auslegung den Fokus der Anwendung darstellt. Die für die Implementierung des finalen Modells genutzte Parametrierung des Modells aus der scikit-learn-Bibliothek ist in Anhang A.4 gegeben. In dieser Implementierung wird zudem auf eine spezielle Initialisierung der Clusterschwerpunkte zurückgegriffen, die in Abschnitt 6.2.3.4 erläutert wird.

---

[267] Hierauf wird in den Abschnitten 6.2.2.1 - 6.2.2.4 eingegangen.

[268] Vgl. scikit-learn: sklearn.cluster.KMeans (2019).

[269] Vgl. Trevino, A.: Introduction to K-means Clustering (2016).

[270] Vgl. scikit-learn: 2.3. Clustering — scikit-learn 0.20.3 documentation (2019).

Für eine Fahrstildetektion ist jedem dieser Cluster die Repräsentanz eines Fahrstils zuzuordnen. Da die Akzeptanzwahrscheinlichkeit einer Zeitlücke im Gegenverkehr beim Linksabbiegen vom Fahrstil abhängt,[271] bietet die Auswertung der Cluster hinsichtlich dieser Akzeptanzwahrscheinlichkeit doppelten Nutzen. Einerseits ist es damit möglich, die Cluster hinsichtlich der Repräsentanz eines Fahrstils zu interpretieren und hierdurch funktionale Anforderungen zu überprüfen. Andererseits ist eine zusätzliche Evaluation des Clusterergebnisses möglich, da der Verlauf der Akzeptanzkurven bzw. deren Trennbarkeit und Überschneidung als Gütemaß des Clusterings dient. Für jeden Datenpunkt liegen Informationen vor, welche Zeitlücke vom zugehörigen Fahrer in der betreffenden Runde, in der der Datenpunkt aufgenommen wurde, akzeptiert wurde. Diese Informationen werden für die im Cluster befindlichen Datenpunkte gemeinsam in einer Kurve dargestellt.

Mit Abbildung 6-5 werden die Akzeptanzkurven der drei unterschiedlichen Cluster dargestellt. Die Akzeptanzkurven der Cluster unterscheiden sich im Beginn der Steigung sowie dem Steigungsgradienten, was auf die Repräsentanz von unterschiedlichen Fahrstilen durch die Cluster hinweist, da die Verläufe qualitativ aus der Literatur bekannten Lückenakzeptanzkurven übereinstimmen.[272] Eine geringere Lückenakzeptanz bzw. Akzeptanzwahrscheinlichkeit bei gleicher Zeitlücke weist auf einen vorsichtigeren Fahrstil hin. C0 wird daher im Folgenden als vorsichtiger, C1 als ausgeglichener und C2 als sportlicher Fahrstil interpretiert.

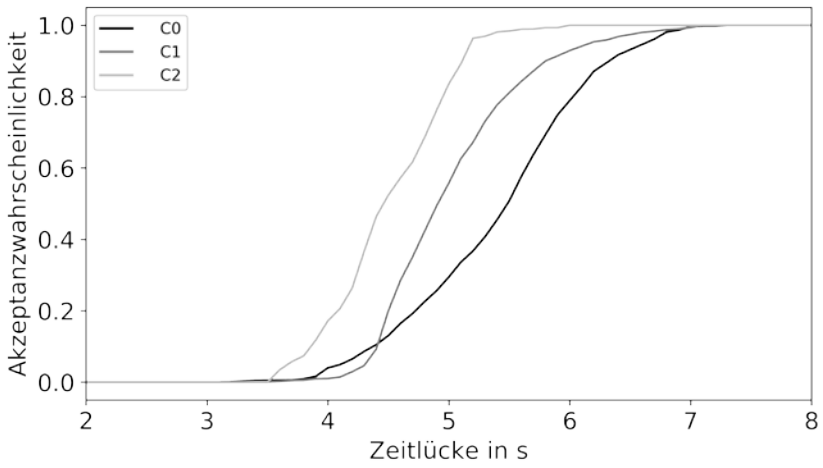


Abbildung 6-5: Akzeptanzkurven für drei unterschiedliche Fahrstile (Links-Abbiegen)

6.2.2 Funktionale Anforderungen

Zur Überprüfung der funktionalen Anforderungen sind diese zunächst aufzustellen. Hierzu wurden verschiedene Veröffentlichungen zum Stand der Technik in der Fahrerklassifizie-

[271] Vgl. Winter, J. C.F. de et al.: Left turn gap acceptance in a simulator (2010), S. 10.

[272] Vgl. Ragland, D. R. et al.: Gap acceptance for vehicles turning left across on-coming traffic, S. 21.

rung[273] [274] [275] hinsichtlich der zusammengefassten Quellen und den hierin identifizierten Zusammenhängen zur Detektion von unterschiedlichen Fahrstilen analysiert, um, bei Übertragbarkeit der Zusammenhänge auf das Linksabbiege-Manöver und die darin verwendeten Eingangsgrößen (nicht unbedingt die verwendeten statistischen Größen), Anforderungen für das gelernte Fahrstilmodell zu extrahieren. Zusätzlich zu den hierdurch aufgestellten Anforderungen wurde eine gezielte Recherche nach den bisher nicht abgedeckten Eingangsgrößen des gelernten Modells durchgeführt, so dass jede der Eingangsgrößen

- Geschwindigkeit
- Längsbeschleunigung
- Querbeschleunigung
- Lenkgeschwindigkeit
- Ruck

mindestens durch eine Anforderung adressiert wird. Da diese minimal geforderte Abdeckung jedes Merkmals im Anwendungsfall erreicht wurde, ist keine weitere Diskussion notwendig, die die Sensitivität eines bisher nicht in der Literatur identifizierten Merkmals thematisiert. Die extrahierten Anforderungen lauten wie folgt:

- L1: Der ordinale Zusammenhang zwischen Längsbeschleunigung bzw. Geschwindigkeit entspricht den aus der Literatur bekannten Beziehungen hinsichtlich der Eingruppierung von unterschiedlichen Fahrstilen (siehe Abbildung 6-6).

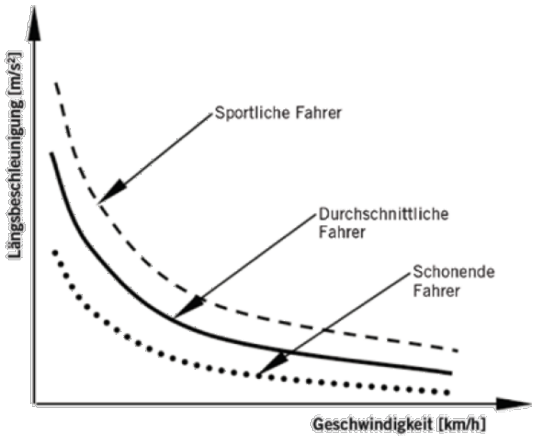


Abbildung 6-6: Anforderung L1[276]

[273] Marina Martinez, C. et al.: Driving Style Recognition for Intelligent Vehicle Control (2018).

[274] Wang, R.; Lukic, S. M.: Review of driving conditions prediction and driving style recognition (2011).

[275] Wang, W. et al.: Modeling and Recognizing Driver Behavior Based on Driving Data (2014).

[276] Bossdorf-Zimmer, J. et al.: Fingerprint des Fahrers (2011), S. 228.

- L2: Die ordinale Gruppierung der Fahrstile zeigt in der Darstellung der Längsbeschleunigung über der Querbeschleunigung eine Verteilung analog zu bekannten Klassifizierungen (siehe Abbildung 6-7).

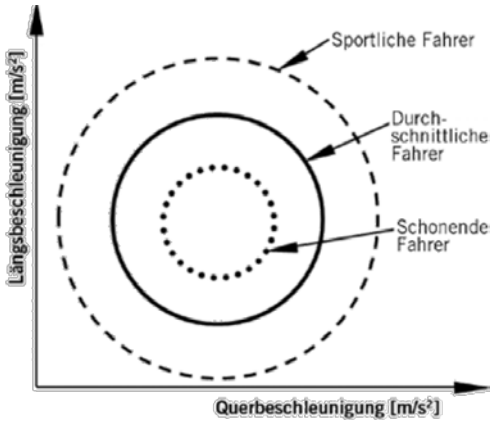


Abbildung 6-7: Anforderung L2[276]

- L3: Eine höhere Lenkradwinkelgeschwindigkeit weist auf einen „aggressiveren“[277] Fahrer hin.
- L4: Ein höherer Ruck weist auf einen „aggressiveren“[278] Fahrstil hin.

Im Folgenden wird auf die Überprüfung der Anforderungen L1 bis L4 in einzelnen Abschnitten eingegangen.

6.2.2.1 Anforderung L1

- L1: Der ordinale Zusammenhang zwischen Längsbeschleunigung bzw. Geschwindigkeit entspricht den aus der Literatur bekannten Beziehungen hinsichtlich der Eingruppierung von unterschiedlichen Fahrstilen (siehe Abbildung 6-6).

Zur Überprüfung dieser Anforderung werden die Maximalwerte der Größen Längsbeschleunigung und Geschwindigkeit der einzelnen Datenpunkte übereinander dargestellt und die aus dem Clustering resultierende Zuordnung der Datenpunkte mit der Soll-Zuordnung des Literatur-Diagramms verglichen. Mit Abbildung 6-8 ist diese Darstellung gegeben.[279] Die Bezeichnung „ausgeglichen“ als Fahrstil in Abbildung 6-8 ist qualitativ mit dem „durchschnittlichen“ Fahrer in Abbildung 6-6 und die Bezeichnung „vorsichtig“ mit dem „schonenden“ Fahrer gleichzusetzen. Die exakte Benennung bzw. Bedeutung der einzelnen Fahrstile ist hierbei nicht von Relevanz, sondern lediglich die mit ihnen verbundene qualitative Reihenfolge ihrer Risikobereitschaft um einen Vergleich zwischen den Zusam-

[277] Carmona, J. et al.: Analysis of Aggressive Driver Behaviour using Data Fusion (2016), S. 89.

[278] Murphey, Y. L. et al.: Driver's style classification using jerk analysis (2009), S. 25.

[279] Die “Lücken” innerhalb der Längsbeschleunigungswerte in dieser und allen folgenden Abbildung ergeben sich aus der Vorverarbeitung der Fahrzeugsensordaten durch das Messsystem.

menhängen im Clustering-Modell und den Literaturangaben herzustellen (sportlich > ausgeglichen/ durchschnittlich > vorsichtig/ schonend).

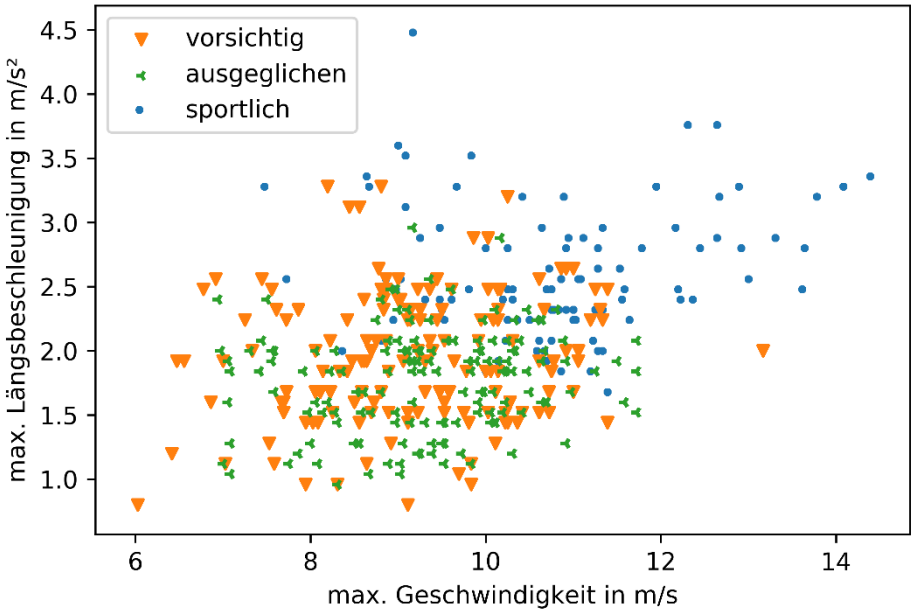


Abbildung 6-8: Überprüfung der Anforderung L1 (originales Modell)

Der Vergleich der Grenzmuster aus der Literatur mit dem Clustering-Ergebnis des Fahrstilmodells zeigt, dass die generelle relative Lage sowie Trennung des sportlichen Fahrstils von den anderen beiden ebenfalls im gelernten Modell auftritt. Eine klare Trennung sowie die in der Literatur vorhandenen Grenzmuster der Cluster „vorsichtig“ und „ausgeglichen“ ist jedoch nicht erkennbar. Ein möglicher Grund ist, dass das Manövermodell nur einen sehr begrenzten Bereich von Beschleunigungsmanövern aufweist, während das Literaturdiagramm auf allgemeinem Fahrten mit einem breiten Spektrum an Manövern zum Beschleunigen und Bremsen basiert.[280] Darüber hinaus ist es möglich, dass zu wenige Daten für das Modell in den Grenzbereichen vorhanden sind, um die Muster deutlich auszubilden. Die generelle Ordinalität der Fahrstile des Cluster-Modells widerspricht jedoch zumindest nicht den in der Literatur angegebenen Zusammenhängen, was aufgrund der differierenden Grundlagen zur Erhebung der Daten in diesem Fall genügt. Hierdurch ist die Anforderung L1 erfüllt.

Dass dies nicht in jeder Konfiguration des Modells der Fall ist, wurde durch die Überprüfung der Anforderung von vorherigen bzw. nicht-finalen Modellständen ermittelt. Enthält das Modell nur die sieben Merkmale der direkt gemessenen Größen des Datensatzes und wird der Ruck nicht als Eingangsmerkmal verwendet, wird der in Abbildung 6-9 dargestellte Zusammenhang zwischen Längsbeschleunigung, Geschwindigkeit und Fahrstil erlernt. Die Ordinalität der Fahrstile widerspricht dem Literatur-Zusammenhang deutlich, da die Datenpunkte, die laut Literatur im „sportlichen“ Bereich eines Fahrstils liegen sollten,

[280] Vgl. Bossdorf-Zimmer, J. et al.: Fingerprint des Fahrers (2011), S. 227.

nicht dem Cluster mit dem höchsten Risikobewusstsein entsprechen. Hierdurch wird ebenfalls das übergeordnete Sicherheitsziel, dass dem Fahrer keinen für ihn zu kleinen Lückenschwellwert zugeordnet wird, verletzt, da die Zuordnung von ausgeglichenen oder vorsichtigen Fahrern fälschlicherweise zu „sportlich“ stattfindet und diesem sportlichen Cluster (C2) kleinere Zeitlücken bei gleicher Akzeptanzwahrscheinlichkeit zugeordnet werden (siehe Abbildung 6-10). Da sich aus diesem Diagramm auch der Schwellwert der Zeitlückenempfehlung für den Fahrstil ergibt, werden dem Fahrer daher zu kleine Zeitlücken empfohlen.

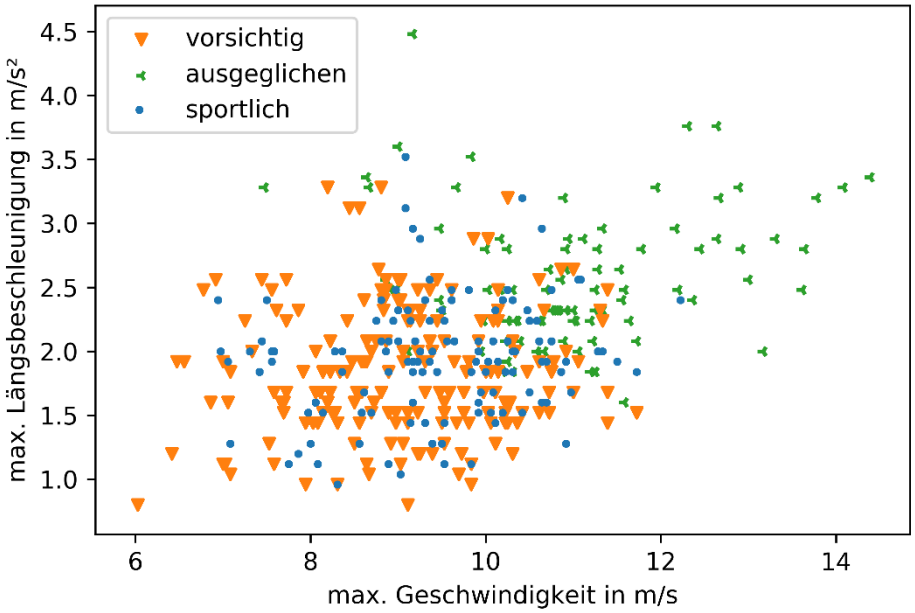


Abbildung 6-9: Überprüfung der Anforderung L1 (Modell ohne Ruck-Eingangsgröße)

Abbildung 6-10: Akzeptanzkurven des Modells ohne Ruck-Eingangsgröße

Das Beispiel zeigt, wie durch die Überprüfung funktionaler Anforderungen Fehlverhalten identifiziert wird, auch wenn dieses Fehlverhalten nicht durch das Vorliegen der Bewer-

tung des Algorithmus ersichtlich wird. Denn auch mit den in Abbildung 6-10 dargestellten Kurven der Modellkonfiguration ohne Verwendung des Rucks wird ein relativ plausibler Verlauf der Akzeptanzwahrscheinlichkeit erreicht. Wenn kein Wissen vorhanden ist, dass mit Hinzunahme der Ruckmerkmale ein besseres Ergebnis hinsichtlich der Trennung sowie der Überlappung der Akzeptanzkurven möglich ist, dann ist auch das Evaluationsergebnis des Modells ohne Ruck ausreichend hinsichtlich der Funktionalität des Modells. Durch die Überprüfung dieser funktionalen Anforderung wird dieses Fehlverhalten jedoch offenbart.

6.2.2.2 Anforderung L2

- L2: Die ordinale Gruppierung der Fahrstile zeigt in der Darstellung der Längsbeschleunigung über der Querbeschleunigung eine Verteilung analog zu bekannten Klassifizierungen (siehe Abbildung 6-7).

Da es sich bei Anforderung L2 ebenfalls um eine Anforderung handelt, die wie L1 eine grafische qualitative Überprüfung der Datenzuordnung in den definierten Dimensionen erfordert, wird die Clusterzuordnung in der Darstellung der Maximalwerte der Längs- über Querbeschleunigung abgebildet. Das Ergebnis ist in Abbildung 6-11 dargestellt. Zur besseren Vergleichbarkeit der Anforderung L2 hinsichtlich der Unterscheidung der einzelnen Fahrstile durch den Kreisdurchmesser ihrer Datenpunkte sind die einzelnen Kreisbögen, die durch die unterschiedlichen Cluster beschrieben werden, in Abbildung 6-12 eingezeichnet sowie die Skalierung der Achsen des Diagramms für eine Kreisdarstellung angepasst.

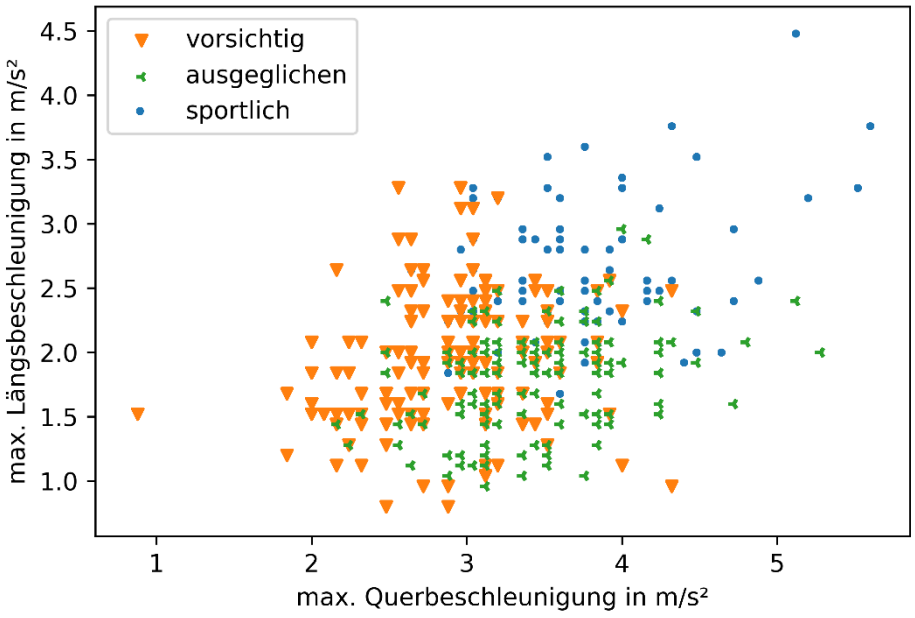


Abbildung 6-11: Überprüfung der Anforderung L2 (originales Modell)

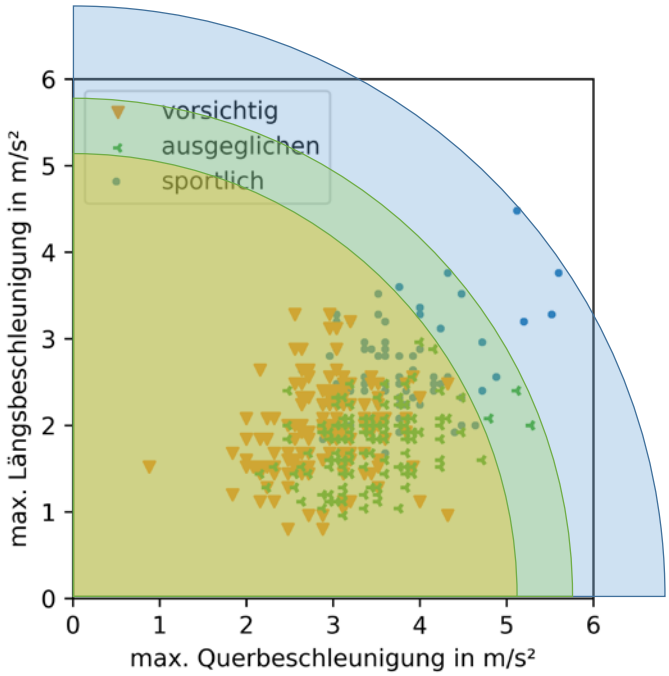


Abbildung 6-12: Vergleich der Anforderung L2 (originales Modell)

Dadurch, dass lediglich die Maximalwerte der Quer- und Längsbeschleunigung zur Überprüfung der Anforderung genutzt werden, befinden sich alle Datenpunkte in einem positiven Wertebereich beider Größen, da das Linksabbiegemanöver aus dem Stillstand eine positive Beschleunigung erfordert. Da die der Anforderung zugrundeliegende Literaturquelle[281] von einem positiven und negativen Wertebereich ausgeht, ist zur Überprüfung ein Viertelkreissegment heranzuziehen. Wie dargestellt lassen sich die unterschiedlichen Fahrstile hinsichtlich der hiermit verknüpften Risikobereitschaft in Kreissegmente mit wachsendem Radius gliedern, was der Anforderung L2 entspricht.

Im Vergleich dazu wird auch die Anforderung L2 durch die Konfiguration des Fahrstilmodells ohne die Merkmale des Rucks verletzt, wie in Abbildung 6-13 dargestellt. Wie bereits im Rahmen der Überprüfung der Anforderung L1 aufgetreten, befindet sich das Cluster der ausgeglichenen Fahrer im Wertebereich teilweise oberhalb dem der sportlichen Fahrer und hat zur Anforderungserfüllung jedoch unter diesem zu liegen. Im direkten Vergleich der Abbildung 6-12 und Abbildung 6-13 wird deutlich, dass viele Datenpunkte im Bereich von 3 - 4 m/s² in der finalen und bisher als funktional korrekt definierten Konfiguration des Modells als „ausgeglichen“ geclustert werden, wohingegen die Konfiguration ohne Ruck diese Datenpunkte als „sportlich“ kennzeichnet. Wie bereits im Rahmen der Anforderung L1 festgestellt, wird hierdurch das übergeordnete Sicherheitsziel ebenfalls verletzt.[282]

[281] Bossdorf-Zimmer, J. et al.: Fingerprint des Fahrers (2011), S. 288.

[282] Siehe Abschnitt 6.2.2.1.

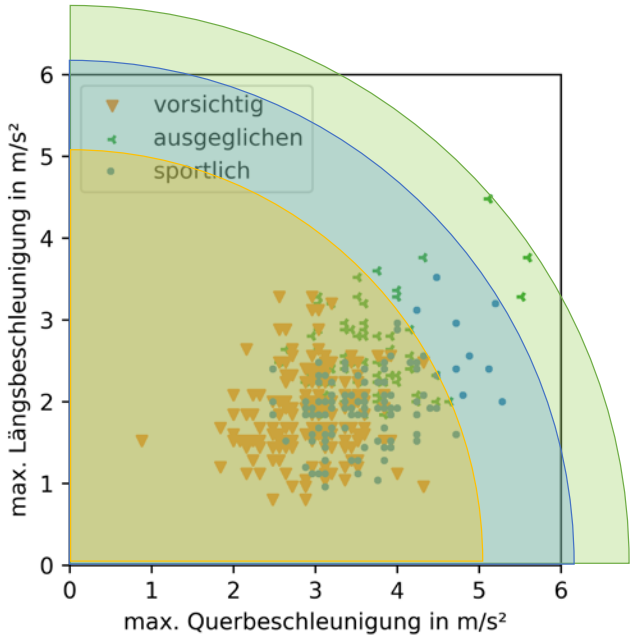


Abbildung 6-13: Vergleich der Anforderung L2 (Modell ohne Ruck-Eingangsgröße)

6.2.2.3 Anforderung L3

- L3: Eine höhere Lenkradwinkelgeschwindigkeit weist auf einen „aggressiveren“ Fahrer hin.

Die Anforderung L3 wird anhand der Darstellung der Häufigkeit eines Werts der maximalen Lenkradwinkelgeschwindigkeit innerhalb der einzelnen Cluster überprüft. Hierzu eignet sich eine kumulative Verteilungsfunktion[283] (CDF), da der Nachteil der Darstellung einer absoluten Quantifizierung der einzelnen, sich in der Menge der zugehörigen Datenpunkten unterscheidenden, Cluster nicht vorliegt. Abbildung 6-14 stellt diese Verteilungsfunktion dar. Wie in L3 gefordert, treten in der Reihenfolge ihrer Aggressivität[284] bzw. Sportlichkeit wachsende Häufigkeiten der Fahrstile bei höheren Lenkradwinkelgeschwindigkeiten auf. Lediglich bis 5 °/s liegen die kumulierte Häufigkeit der Lenkradwinkelgeschwindigkeit des ausgeglichenen Clusters und des sportlichen Clusters zusammen. Dies wird jedoch aufgrund der sonstigen Übereinstimmung der Anforderung und des plausiblen Verlaufs der Häufigkeitsverteilungen als Ausreiser bewertet, wodurch die Anforderung L3 als durch das finale Modell der Fahrstildetektion des Manövers Linksabbiegen erfüllt angesehen wird.

[283] Englische Bezeichnung: cumulative distribution function.

[284] Der Begriff der Aggressivität wird in der zugrundeliegenden Literaturquelle genutzt. Da dieser Begriff jedoch mit Verhaltensweisen, die sich gegen andere Verkehrsteilnehmer richten, konnotiert ist, wird in der vorliegenden Arbeit von „Sportlichkeit“ gesprochen.

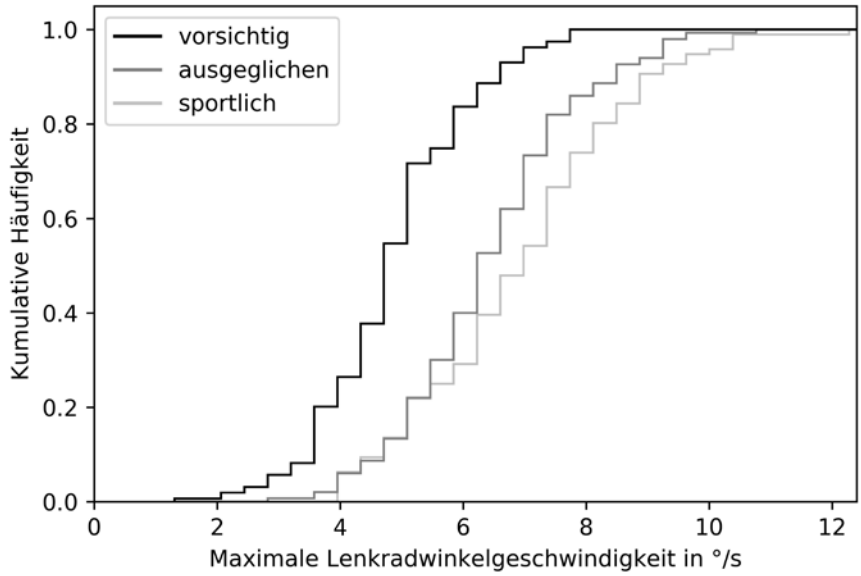


Abbildung 6-14: Überprüfung der Anforderung L3 (originales Modell)

Der Vergleich der Clusterzuordnung der Modellkonfiguration ohne die Nutzung des Merkmals Ruck zeigt, wie bereits in L1 und L2, dass die funktionale Anforderung durch diese Konfiguration nicht erfüllt wird. Die Häufigkeit der Wertebereiche der aufgetretenen Lenkradwinkelgeschwindigkeiten des als sportlich identifizierten Clusters liegen unter der des ausgeglichenen Clusters (siehe Abbildung 6-15). Hierdurch wird ebenfalls (wie bereits bei L1 und L2 aufgefallen) das übergeordnete Sicherheitsziel verletzt.

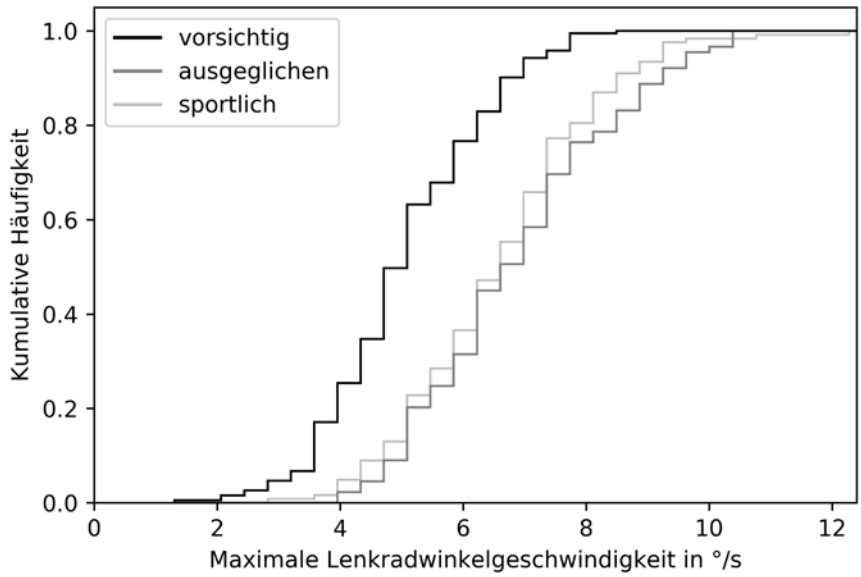


Abbildung 6-15: Überprüfung der Anforderung L3 (Modell ohne Ruck-Eingangsgröße)

6.2.2.4 Anforderung L4

- L4: Ein höherer Ruck weist auf einen „aggressiveren“ Fahrstil hin.

Die Überprüfung der Anforderung L4 findet, wie auch die der Anforderung L3, durch die Darstellung der Häufigkeit der Ruckwerte innerhalb der einzelnen Cluster in der CDF-Darstellung statt. Die maximalen Ruckwerte werden dabei separat von den minimalen Ruckwerten betrachtet, da es sich um jeweils um getrennte Handlungen des Fahrers handelt.[285] Die Anforderung L4 ist für beide Extremwerte anwendbar, da die ihr zugrundeliegende Literaturquelle ebenfalls Extremwerte im positiven und negativen Wertebereich betrachtet.[286]

Die CDF-Darstellung des maximalen Rucks für die drei Cluster ist mit Abbildung 6-16 gegeben, die für den minimalen Ruck in Abbildung 6-17. Die kumulative Häufigkeit des vorsichtigen Clusters liegt im gesamten Wertebereich unter der des ausgeglichenen Clusters, wodurch die Anforderung L4 verletzt wird. Dabei ist die Trennung der beiden Cluster hinsichtlich der kumulativen Häufigkeit deutlich, wodurch die Argumentation mit einzelnen Ausreisern hier nicht gültig ist. Das sportliche Cluster hingegen entspricht der Anforderung, da dieser als Äquivalent zum aggressivsten Fahrstil die meisten Ruckwerte im hohen Wertebereich besitzt.

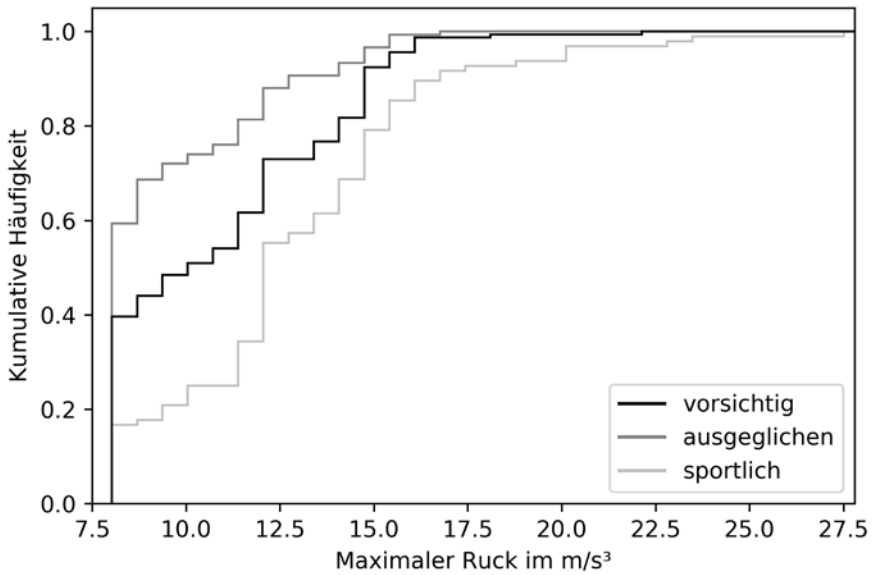


Abbildung 6-16: Überprüfung der Anforderung L4 (max. Ruck, originales Modell)

Auch die Auswertung des minimalen Rucks zeigt eine Verletzung der Anforderung L4 durch die inkorrekte Reihenfolge der Cluster vorsichtig und ausgeglichen. Das vorsichtige Cluster hat zur Erfüllung der Anforderung die betragsmäßig niedrigsten Werte am häufigsten aufzuweisen, was jedoch im gelernten Modell das ausgeglichene Cluster übernimmt.

[285] Siehe Abschnitt 6.2.1.

[286] Vgl. Murphey, Y. L. et al.: Driver's style classification using jerk analysis (2009), S. 24 f.

Das sportliche Cluster erfüllt auch in diesem Fall die Anforderung, abgesehen von der restlichen Clusterreihenfolge. Insgesamt ist die Anforderung L4 sowohl für positive als auch negative Ruckwerte nicht erfüllt, wodurch das übergeordnete Sicherheitsziel, dass der Fahrer keinen für ihn zu kleinen Lückenschwellwert erhält, verletzt wird.

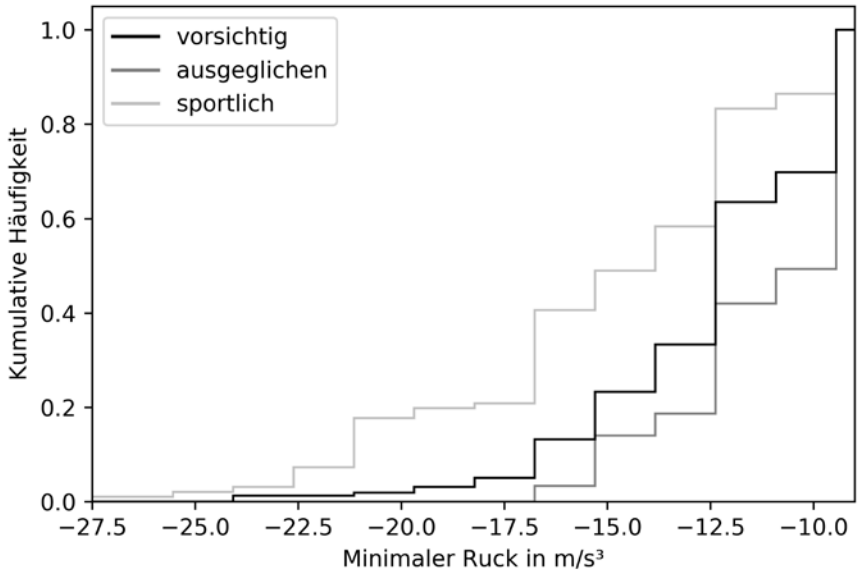


Abbildung 6-17: Überprüfung der Anforderung L4 (min. Ruck, originales Modell)

Dadurch, dass die übrigen Anforderungen durch das gelernte Modell in dieser Konfiguration jedoch erfüllt werden, wird die Anforderung L4 hinsichtlich ihrer Anwendbarkeit und Erhebungsgrundlage detaillierter untersucht. In der dieser Anforderung zugrundeliegenden Literaturquelle[287] beschränkt sich die Aussage, dass ein höherer Ruck auf Fahrstile mit einer höheren Aggressivität hinweist, auf Abschnitte mit gleicher Straßenkategorisierung und gleich hohem Verkehrsaufkommen. Dies wird damit begründet, dass beispielsweise ein aggressiver Fahrer bei geringem Verkehrsaufkommen auf einer Autobahn geringere Beschleunigungswerte aufweist als der gleiche Fahrer bei einem hohen Verkehrsaufkommen.[286] Dieser Voraussetzung des gleichen Verkehrsaufkommens und der gleichen Straßenkategorie wird jedoch auch der vorliegende Anwendungsfall bzw. zugrundeliegende Datensatz gerecht, da es sich um das gleiche Linksabbiegemanöver an der gleichen Kreuzung handelt. Auch Unterschiede des Beschleunigungsvermögens der in den Clustern befindlichen Datenpunkten sind auszuschließen, da das gleiche Fahrzeug zur Erhebung aller Daten genutzt wurde. Jedoch besitzt dieses Fahrzeug einen durch das Automatikschaltgetriebe verursachten relativ hohen Anfahrruck, was viele Probanden gerade in der Eingewöhnungsstrecke, welche nicht in den Datensatz aufgenommen wurde, überraschte. Eine Datenanalyse hinsichtlich der Häufigkeit der Fahrerfahrung der drei Cluster zeigt, dass sich Datenpunkte von Fahrern mit einer jährlichen Kilometeranzahl von kleiner gleich 10.000 km am häufigsten im Cluster „vorsichtig“ befinden (siehe Abbildung 6-18). Die Bewertung hinsichtlich der jährlich gefahrenen Kilometer wurde gewählt, da die Annahme be-

[287] Murphey, Y. L. et al.: Driver's style classification using jerk analysis (2009).

steht, dass die Kontrollierbarkeit des Automatikgetriebes stärker mit dem aktuellen „Übungszustands“ des Fahrers zusammenhängt. Eine Gesamtfahrleistung, deren Verteilung nicht genauer bekannt ist und könnte auch aus Fahrten, die weit in der Vergangenheit liegen, stammen.

Abbildung 6-18: Verteilung der Datenpunkte von Fahrern geringer Fahrerfahrung

Dieser Umstand in Kombination mit dem Wissen über den hohen Anfahrruck des Fahrzeugs erklärt die im Vergleich zur Anforderung L4 veränderte Reihenfolge der Cluster hinsichtlich der mit ihnen verbundenen Aggressivität bzw. Sportlichkeit bei der Betrachtung des maximalen Rucks. Die erfahreneren Fahrer konnten das untypische Anfahrverhalten des Testfahrzeugs besser kontrollieren und verursachten hierdurch geringere Ruckwerte. Die der Anforderung L4 zugrundeliegende Literaturquelle[287] trifft keine Aussage darüber, welche Fahrerfahrung die einzelnen Fahrer innerhalb des verwendeten Datensatzes besitzen bzw. ob dieser Einflussfaktor berücksichtigt wurde. Daher wird die Anforderung L4 für diesen speziellen Fall als nicht gültig beurteilt. Auch die Abweichung des Merkmals „minimaler Ruck“ ist durch die Verteilung der Fahrerfahrung innerhalb der Cluster erklärbar.

An diesem Beispiel wird deutlich, welche Vorteile die Anwendung von ML gegenüber konventioneller Programmierung besitzt. Durch Implementierung der Anforderung L4, wie es im Rahmen einer konventionellen Programmierung vorgesehen ist, und deren darauffolgende Parametrierung an Realdaten würde ein funktional falsches Verhalten des resultierenden Modells resultieren, da diese Anforderung aufgrund des Kontextes (hier: der Versuchsträger) nicht gültig ist. Zusätzlich wird hierdurch die Forderung von Salay und Czarnecki[288], dass funktionale Anforderungen als „Vorwissen“ vor dem Training des gelernten Modells zu nutzen sind (siehe Abschnitt 3.2.2.3), konkretisiert. Es sind lediglich die Anforderungen als „Vorwissen“ einzubeziehen, deren Anwendbarkeit und Korrektheit im Vorfeld in Bezug auf den Anwendungskontext sichergestellt sind, da sonst der Vorteil des ML nicht genutzt wird.

[288] Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018).

6.2.3 Robustheitsanforderungen

Zur Überprüfung der in Unterkapitel 5.5 aufgestellten Robustheitsanforderungen sind diese zunächst auf ihre Relevanz bzw. Gültigkeit für die genutzte Methode des ML im Anwendungsfall zu untersuchen. Die relevanten Anforderungen werden anschließend auf das Modell der Fahrstildetektion beim Manöver Linksabbiegen angewendet.

Durch die Nutzung des K-Means-Algorithmus, welcher alle zur Verfügung stehenden Daten gesammelt zum Training nutzt, besitzt die Reihenfolge der Daten im Datensatz keine Auswirkungen auf das Modellergebnis.[289] Hierdurch wird die Anforderung T2 „Veränderungen der Datensequenzen während des Trainingsprozesses besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells“ immer erfüllt, weshalb sie keiner expliziten Überprüfung bedarf. Die übrigen Anforderungen sind in der Lage das Modellergebnis zu beeinflussen, weshalb sie im Folgenden entsprechend der in Unterkapitel 5.5 vorgenommenen Kategorisierung überprüft werden, um die Anwendbarkeit dieses Schrittes zur Identifikation fehlender Generalisierbarkeit zu untersuchen. Dabei besteht, wie bereits erwähnt, keine Notwendigkeit nach vollständiger Erfüllung der Anforderungen. Sie dienen stattdessen dazu, Schlüsse auf die Generalisierbarkeit des Modells zu ziehen und hierdurch ein passendes Sicherheitskonzept für das Modell, wie beispielsweise eine Beschränkung des Betriebsbereichs des gelernten Modells, zu erarbeiten.

6.2.3.1 Datenquantität DQ1 und DQ2

Es wird zunächst auf die Anforderungserfüllung durch das Modell in jeder der beiden Anforderungen separat eingegangen, um anschließend die Anwendbarkeit der Anforderungen zu diskutieren und ein Fazit zu den Erkenntnissen über die Generalisierbarkeit des final gelernten Modells aus diesen Anforderungen zu ziehen.

Anforderung DQ1

- DQ1: Die gleichmäßige Änderung (bspw. über alle Klassen hinweg) der Datensatzgröße in den unterschiedlichen Entwicklungsphasen bis zu einem gewissen Schwellwert besitzt keine Auswirkungen auf die grundlegende Funktionalität des Modells.

Diese Anforderung wird durch die Auswahl einer Untermenge an Daten aus dem originalen Datensatz und dem anschließenden Training eines Algorithmus mit der gleichen Konfiguration wie das originale Modell auf dieser Untermenge überprüft. Anschließend wird dieses gelernte, reduzierte Modell auf die verbleibenden, ungesehenen Daten des originalen Datensatzes angewendet. Die hieraus resultierenden Clusterzuordnungen werden mit denen des ursprünglichen Modells (dem „originalen“ oder „finalen“ Modell), welches auf dem vollen Datensatz trainiert wurde, verglichen. Eine geringe Übereinstimmung der Mo-

[289] Siehe Abschnitt 6.2.1.

dellergebnisse weist auf ein Modell hin, welches eine hohe Sensitivität hinsichtlich der Datenquantität besitzt. Neben der Clusterübereinstimmung wird ebenfalls die Erfüllung der funktionalen Anforderungen L1 bis L3 als Gütekriterium des neu trainierten Modells hinzugezogen, da deren Erfüllung ein Indiz für das funktional korrekte Verhalten darstellen. Die einzelnen Testfälle werden in Tabelle 6-1 zusammengefasst.

Tabelle 6-1: Testfälle der Anforderung DQ1

Nr.	**Übereinstimmung aller Cluster**	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Reduzierung auf …**	**Auswahl der Untermenge**
1	87 %	ja	ja	ja	67 %	Zufall
2	45 %	nein	nein	ja	67 %	Zufall
3	97 %	ja	ja	ja	67 %	Zufall
4	42 %	nein	nein	diskutabel	67 %	weniger Fahrer
5	95 %	ja	ja	ja	67 %	weniger Fahrer
6	96 %	ja	ja	ja	67 %	weniger Fahrten
7	95 %	ja	ja	ja	67 %	weniger Fahrten
8	94 %	ja	ja	ja	67 %	weniger Fahrten
9	97 %	ja	ja	ja	67 %	weniger Fahrten
10	95 %	ja	ja	ja	67 %	weniger Fahrten
11	43 %	nein	nein	ja	50 %	weniger Fahrten
12	93 %	ja	ja	nein	50 %	weniger Fahrten

Die Auswahl der Untermenge erfolgt unter der Voraussetzung, dass die relative Verteilung der Cluster beibehalten, d.h. jedes Cluster um den gleichen Anteil reduziert wird, da die Auswirkung der Änderung der Klassengrößen in einer separaten Anforderung (DQ2) untersucht wird. Bei der Anwendung des beschriebenen Vorgehens wurden zunächst 67 % der gesamten Daten als Untermenge zum Training eines neuen Modells genutzt, um einen deutlichen Unterschied zwischen den Modellen hinsichtlich der genutzten Datenmenge zu erzielen.

In Testfall Nr. 1 wurde die Untermenge zufällig aus den zur Verfügung stehenden Daten ausgewählt. Dabei wird eine Übereinstimmung der Clusterzuordnung zwischen originalem, mit vollem Datensatz trainierten, und neuem, mit der Untermenge trainierten, Modell von 87 % erreicht. Die funktionalen Anforderungen werden durch das hieraus resultierende Modell erfüllt. Da die Auswahl der zu der Untermenge gehörenden Daten zufällig getroffen wird, werden weitere Durchläufe durchgeführt, um die Reproduzierbarkeit des Ergebnisses zu untersuchen. Im zweiten Testfall, mit einer anderen zufälligen Datenauswahl, wird lediglich eine Übereinstimmung der Clusterzuordnung von 45 % erreicht. Auch die

funktionalen Anforderungen L1 und L2 werden durch das neu trainierte Modell verletzt, was wiederum zeigt, dass deren Überprüfung ein Fehlverhalten aufgrund von einer fehlenden Datenmenge aufdeckt. Die in L3 geforderte Ordinalität der Cluster ist im reduzierten Modell vorhanden. Im dritten Testfall beträgt die Übereinstimmung 97 %. Hieran wird deutlich, welchen großen Einfluss die Auswahl von Daten auf das Modellergebnis besitzt und dass ein reproduzierbares quantitatives Ergebnis dieser Anforderungsüberprüfung nicht mit einer zufälligen Auswahl der Untermenge getroffen wird. Durch die niedrige Übereinstimmung der Cluster im Testfall Nr. 2 ist jedoch die Aussage möglich, dass eine hohe Modellsensitivität auf die Quantität der Daten besteht, wenn die Auswahl der Daten zufällig, auch unter der Voraussetzung der gleichbleibenden Clusterrepräsentanz, um 33 % erfolgt. Es ist möglich, in weiteren Testfällen den Schwellwert zu suchen, ab dem einerseits Reproduzierbarkeit der Vorhersage erreicht und andererseits die Anforderung DQ1 erfüllt wird, jedoch wird der Kenntnis dieses Schwellwerts kein Mehrwert hinsichtlich der Untersuchung der Generalisierbarkeit des Modells zugeschrieben. Dies ist darauf zurückzuführen, dass eine zufällige Entfernung der Datenpunkte keinen im Rahmen der Datenerhebung realistischen Fall der Reduktion des Datensatzes darstellt.

Daher wird die Art der Reduktion der Gesamtdaten zum Training in weiteren Testfällen variiert. Für eine systematische Auswahl werden einerseits die Fahrten einzelner Fahrer gezielt reduziert und andererseits die Daten pro Fahrer. Hierdurch wird im ersten Fall simuliert, dass in der Datenerhebung weniger Fahrer aufgenommen, d.h. der Probandenpool verkleinert wurde, und im zweiten Fall, dass bei gleichbleibend großem Probandenpool pro Proband weniger Fahrten durchgeführt wurden. Mit einer gleichmäßigen Verringerung einzelner Fahrer über alle Cluster hinweg auf weniger als 67 % der Daten, wobei die relative Clusterverteilung zum ursprünglichen Datensatz beibehalten wurde, wird im fünften Testfall eine Übereinstimmung von 42 % erreicht. Die funktionalen Anforderungen L1 und L2 werden durch dieses Modell verletzt. Die Anforderungserfüllung von L3 ist nicht direkt ausgeschlossen, da die Ordinalität der Cluster der Anforderung entsprechen, jedoch zwei der drei Cluster schlecht trennbar sind und sich im oberen Wertebereich überschneiden.[290] Im sechsten Testfall wird, mit einer erneuten gleichmäßigen Reduzierung der Fahrer in allen Clustern, eine Übereinstimmung der Clusterzuordnung von 99 % erreicht, wobei hier die Anforderungen L1 bis L3 erfüllt werden. Auch durch diese Veränderungsmethode ist aufgrund der großen Abweichungen der Ergebnisse keine reproduzierbare Aussage hinsichtlich des quantitativen Einflusses der zum Training genutzten Datenmenge bei gleichmäßiger Datenreduzierung möglich. Das geringe Übereinstimmungsergebnis in Testfall Nr. 5 zeigt jedoch, dass eine hohe Modellsensitivität auf interindividuelle Unterschiede der Probanden bei einer Reduktion um 33 % vorliegt, selbst wenn die relative Clusterverteilung beibehalten wird. Durch diese Art der Datenreduzierung wird die Anforderung DQ1 ebenfalls verletzt.

[290] Siehe Anhang 8-9 in D.1

Der zweite Fall, die gleichmäßige Verminderung der Fahrten pro Fahrer bei gleichbleibender relativer Verteilung der Cluster und gleichbleibenden Fahrerpool, ergab eine Übereinstimmung im Wertebereich von 96 % bis 94 % in fünf Testfällen. Die Datenmenge wird dabei, wie bereits in den anderen Testfällen, bei 67 % der originalen Daten belassen. In allen Fällen werden die funktionalen Anforderungen L1 bis L3 erfüllt, was im Rahmen dieser hohen Übereinstimmung zu erwarten ist. Durch dieses bisher reproduzierbare Ergebnis wird festgestellt, dass der Einfluss der unterschiedlichen Fahrten pro Fahrer auf die im gelernten Modell verorteten Zusammenhänge geringer ist als die Menge an unterschiedlichen Fahrer. Die interindividuellen Unterschiede der Probanden besitzen in diesem Anwendungsfall daher eine höhere Relevanz als die intraindividuellen Unterschiede. Die Modellsensitivität auf intraindividuelle Unterschiede, unter der Voraussetzung der gleichbleibenden Clusterrepräsentanz, wird hierdurch als gering eingestuft. Die Anforderung DQ1 ist für eine Reduzierung der Fahrten um 33 % pro Cluster erfüllt. Für weitere Datenerhebungen in diesem Anwendungsfall ist daher der Fokus auf ein breites Probandenspektrum zu legen anstelle der Erhöhung der Wiederholungen pro Fahrer, um die Qualität des Modells zu erhöhen bzw. eine höhere Generalisierbarkeit zu erreichen. Diese Aussagen besitzen allerdings keine statistische Signifikanz, da der Stichprobenumfang der zufälligen Testfälle zu gering ist. Um beispielsweise mit einer Konfidenz von über 95% statistisch signifikant festzustellen, dass diese Aussagen korrekt sind, sind mindestens 59 Testfälle durchzuführen.[291] Im Rahmen der vorliegenden Arbeit wird auf die tatsächliche Durchführung der Mindesttestfallanzahl für statistische Signifikanz verzichtet, da der Fokus auf der Anwendung bzw. Anwendbarkeit des Ansatzes und den resultierenden Erkenntnissen für Generalisierbarkeit liegt und nicht auf dem tatsächlichen Nachweis der Generalisierbarkeit des gelernten Modells der Fahrstildetektion.

Testfall Nr. 11 und 12 bestehen aus einem komplementären Datensatz, wobei geprüft wurde, ob bei einer Halbierung der Fahrten pro Fahrer bei gleicher Clusterrepräsentanz zwei, im Vergleich zum originalen gelernten Modell, leistungsfähige Modelle trainiert werden. Bei Bestätigung ist eine hohe Sicherheit gegeben, dass die Anzahl der Fahrten pro Fahrer für die Extraktion der relevanten Zusammenhänge im originalen Modell genügen. Das Testergebnis zeigt jedoch, dass durch Training mit einer Hälfte des Datensatzes eine lediglich geringe Übereinstimmung der Clusterergebnisse von 43 % erreicht wird, mit der anderen Datenhälfte 93 %. Die funktionalen Anforderungen werden in keinem der Testfälle vollständig erfüllt. Durch dieses Ergebnis lässt sich daher lediglich ableiten, dass eine bei einer gleichmäßigen Reduktion der Fahrten pro Fahrer um 50 % die Anforderung DQ1 verletzt wird.

Um den Grenzwert zu finden, ab dem die Anforderung DQ1 erfüllt wird, sind weitere Testfälle mit einer veränderten Reduktionsmenge zwischen 50 % und 67 % durchzuführen.

[291] Die Berechnungen und deren Grundlagen finden sich in Anhang D.6. Die Grundlagen der Berechnung gelten ebenfalls für die weiteren Robustheitsanforderungen, die auf einer zufälligen Auswahl an Testfällen basieren.

Jedoch wird auf diese Testfälle verzichtet, da das Kennen dieses Grenzwerts lediglich die Sicherheit gibt, dass genügend Fahrten pro Fahrer im Trainingsdatensatz vorhanden sind, was bereits mit 33 % erfüllt wird.

Anforderung DQ2

- DQ2: Alle beabsichtigten Klassen sind innerhalb des Trainingsdatensatzes für die grundlegende Funktionalität des Modells hinreichend vertreten. Die Veränderung der Klassenrepräsentanz einzelner Klassen verändert die Leistungsfähigkeit für jede andere beabsichtigte Klasse nicht.

Zur Feststellung der Modellsensitivität auf die Quantität der Daten lässt sich neben der Auswirkung der gleichmäßigen Reduktion der Daten bei gleichbleibender Clusterrepräsentanz auch die Auswirkung der Reduktion eines Clusters bei gleichbleibender Datenmenge für die anderen Cluster untersuchen. Hierdurch wird der Fall überprüft, dass durch eine nicht-systematische Auswahl der Probanden ein Cluster stark unterrepräsentiert ist. Es wird erwartet, dass bei einer hohen Generalisierbarkeit des Modells die Trennungsfähigkeit bzw. Zuordnung der anderen Cluster nicht darunter leidet.

Zur Überprüfung wird die Anzahl der einzelnen Cluster durch Zufallsauswahl reduziert und wie bereits im Rahmen der Überprüfung DQ1[292] der Vergleich der Clusterzuordnung des Modells, welches auf dem reduzierten Datensatz trainiert wurde, mit der des originalen Modells durchgeführt. Im Rahmen dieser Anforderungsüberprüfung findet lediglich eine zufällige Auswahl der Datenuntermenge in den Testfällen Anwendung, da im Fall der Reduktion eines einzelnen Clusters keine Begründung einer systematischen Datenauswahl, wie beispielsweise im Rahmen der Überprüfung von DQ1[292] mit der Verkleinerung des gesamten Probandenpools um bestimmte Fahrer, existiert. Dies liegt unter anderem darin begründet, dass sich der Fahrstil eines Fahrers innerhalb der durchgeführten Fahrten (laut Vorhersage des finalen Modells der Fahrstildetektion) zwischen zwei aufeinanderfolgenden Runden der Teststrecke teilweise ändert. Hierdurch ist es nicht möglich einzelne Fahrer aus dem Gesamtdatensatz zu entfernen, da ein Fahrer mehreren Fahrstilen zugeordnet ist.

Zur Evaluation der resultierenden Modelle werden die Cluster miteinander verglichen, deren Anzahl nicht reduziert wurde (Übereinstimmung verbleibende Cluster, ÜvC), um die Anforderung entsprechend ihrer Spezifikation zu untersuchen. Zusätzlich wird die Übereinstimmung aller Cluster (ÜaC) überprüft, um die Verringerung der Gesamtvorhersageleistung festzustellen. Um ein Gütemaß der Funktionalität des Modells zu erhalten, wird ebenfalls die Erfüllung der funktionalen Anforderungen L1 bis L3 untersucht.

Insgesamt werden mehrere Testfälle mit gleicher Konfiguration und unterschiedlicher zufälliger Auswahl der Daten durchgeführt, um die Stärke der Abhängigkeit der Clusterübereinstimmung von der Zufälligkeit der Auswahl festzustellen. Als Ausgangspunkt der An-

[292] Die gleichmäßige Änderung (bspw. über alle Klassen hinweg) der Datensatzgröße in den unterschiedlichen Entwicklungsphasen bis zu einem gewissen Schwellwert besitzt keine Auswirkungen auf die grundlegende Funktionalität des Modells.

forderungsüberprüfung wird eine Reduktion der Datenmenge eines Clusters auf 33 % genutzt, da bei einer solchen hohen Verminderung der Quantität eine Änderung der Clusterübereinstimmung der verbleibenden Cluster zu erwarten ist, falls Zusammenhänge des reduzierten Clusters zur Vorhersage der anderen Cluster genutzt werden. Die Ergebnisse der einzelnen Testfälle sind in Tabelle 6-2 gelistet.

Tabelle 6-2: Testfälle der Anforderung DQ2

Nr.	**Übereinstimmung alle Cluster (ÜaC)**	**Übereinstimmung verbleibende Cluster (ÜvC)**	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Reduzierung auf … (andere Cluster 100 %)**
1	83 %	80 %	ja	ja	nein	33 % C0
2	63 %	55 %	diskutabel	diskutabel	nein	33 % C0
3	82 %	78 %	ja	ja	nein	33 % C0
4	70 %	63 %	diskutabel	diskutabel	nein	33 % C0
5	83%	80 %	ja	ja	nein	33 % C0
6	50 %	66 %	diskutabel	diskutabel	ja	33 % C1
7	55 %	66 %	diskutabel	ja	ja	33 % C1
8	52 %	64 %	diskutabel	ja	ja	33 % C1
9	52 %	69 %	diskutabel	diskutabel	ja	33 % C1
10	59%	66 %	diskutabel	ja	ja	33 % C1
11	47 %	60 %	nein	nein	nein	33 % C2
12	47 %	60 %	nein	nein	nein	33 % C2
13	47 %	60 %	nein	nein	nein	33 % C2
14	40 %	37 %	nein	nein	ja	33 % C2
15	42 %	40 %	nein	nein	ja	33 % C2

Die Reduktion des vorsichtigen Clusters wird in den Testfällen Nr. 1 bis 5 überprüft. Dabei variiert die ÜvC von 55 % bis 80 %. Die hohe Streuung der Ergebnisse zeigt aufgrund der Zufallsauswahl der Daten innerhalb der Testfälle, dass die im Clusters C0 verbleibenden Daten einen hohen Einfluss auf die Zuordnung der anderen Cluster besitzen. Dies wird ebenfalls in der Analyse der Veränderungen der Clustervorhersagen zwischen reduziertem und originalem Modell deutlich. Bei Erfüllung der Anforderung, d.h. wenn C0 die Trennbarkeit der beiden anderen Cluster nicht beeinflusst, treten keine Vorhersagenänderungen zwischen diesen beiden Clustern (C1 und C2) auf. Übergänge zwischen C0 und C1 und zwischen C0 und C2 führen hingegen nicht zu einer Verletzung der Anforderung. In den

Testfällen Nr. 1 bis 5 ändert sich jedoch die Vorhersage von C1 zu C2 zwischen originalen und reduzierten Modell, was den Einfluss des Clusters C0 zur Trennung dieser beiden Cluster zeigt. Die Anforderung L3, die sich auf die Eingangsgröße der Lenkradwinkelgeschwindigkeit bezieht, wird in keinem Testfall erfüllt, da die Ordinalität der nicht-reduzierten Cluster vertauscht ist und zusätzlich eine schlechte Trennbarkeit von zwei der drei Cluster in der CDF-Darstellung vorliegt.[293] Die Eingangsgröße der Lenkradwinkelgeschwindigkeit ist aufgrund dieses Ergebnisses detaillierter hinsichtlich ihrer Relevanz zu untersuchen, da durch die Reduzierung eines Clusters zumindest die Trennbarkeit der CDF-Verläufe der nicht-reduzierten Cluster vorzuliegen hat, was in Testfall Nr. 1, 3 und 5 nicht der Fall ist. Durch die hohe generelle Streuung der Ergebnisse, die Übergänge der Clustervorhersagen zwischen den beiden nicht-reduzierten Clustern und das eindeutige Verletzen der Anforderung L3 im Testfall 2, wird die Anforderung DQ2 bei einer Verminderung des Clusters C0 auf 33 % der vorherigen Datenmenge nicht erfüllt.

In den Testfällen Nr. 6 – 10 werden die Ergebnisse der Modelle, die auf einem Datensatz mit 33% der ursprünglichen Größe des ausgeglichenen Clusters (C1) trainiert wurden, untersucht. Die Ergebnisse der ÜvC besitzen in den fünf Testfällen eine Streuung von 5 % und eine ähnliche Erfüllung der Anforderungen L1 bis L3 innerhalb der Testfälle, weshalb von reproduzierbaren Ergebnissen ausgegangen wird. Abgeleitet hieraus wird, dass das finale Modell eine geringere Sensitivität auf eine zufällige Datenauswahl des Clusters „ausgeglichen“ besitzt als bei der Auswahl der Untermenge des Clusters „vorsichtig“. Dies weist entweder darauf hin, dass, unabhängig der Auswahl der Daten des Clusters „ausgeglichen“, die gleichen Zusammenhänge zur Trennung der anderen Cluster erlernt werden. Oder es weist darauf hin, dass die Übereinstimmung von 64% der nicht-reduzierten Cluster resultiert, wenn keine Zusammenhänge des Clusters C1 zur Trennung der anderen Cluster genutzt werden. Eine detaillierte Analyse der veränderten Zuordnungen zeigt, dass es keine Veränderung der Vorhersage von Datenpunkten des Clusters C0 nach C2 oder von Datenpunkten des Clusters C2 nach C0 stattfindet, was die Vermutung, dass durch die Reduktion die Trennbarkeit der anderen Cluster voneinander nicht berührt wird, bestätigt. Daneben zeigen die erzielten Ergebnisse, dass die Zusammenhänge zur Abgrenzung des Clusters C1 im Vergleich zu C0 robust gegenüber der Auswahl der Daten ist, da die ÜaC zwischen reduziertem und originalen Modell ebenfalls eine relativ gesehen geringe Streuung von 9 % aufweist. Die ÜaC ist in allen Testfällen niedriger als die ÜvC. Da sich beide Werte durch die fehlende Übereinstimmung des reduzierten Clusters, hier C1, unterscheiden, zeigt sich, dass die Leistungsfähigkeit der Vorhersage des Clusters C1 stark vermindert ist. Die Anforderung DQ2 wird im Fall der Reduktion des Clusters „ausgeglichen“ auf 33 % nicht verletzt, da keine Übergänge der Datenpunkte zwischen C0 und C2 stattfinden.

Die Verminderung des Clusters C2 („sportlich“) in den Testfällen Nr. 11 – 15 resultiert in einer ÜvC von 37 bis 60 %, was analog zur Streuung der Ergebnisse der Testfällen Nr. 1 – 5 die Sensitivität auf die zufällige Auswahl der Untermenge an vorhandenen Daten des

[293] Siehe Anhang 8-10 bis Anhang 8-14 in D.2

Clusters C2 zeigt. Die in Cluster C2 verorteten Zusammenhänge dienen daher zur Trennung der Cluster C1 und C2. In Testfall Nr. 14 findet beispielsweise ein Übergang von knapp der Hälfte der Vorhersagenänderungen von reduziertem zu originalen Modell zwischen Cluster C0 und C1 statt. Jedoch ist die Streuung in der ÜaC von 7 % im Gegensatz hierzu gering, was vermuten lässt, dass die innerhalb der 33 % der Daten vorhandenen Zusammenhänge des Clusters C2 (im Vergleich zu C1) robust sind und in jeder Zufallsauswahl auftreten. Allerdings ist diese Aussage aufgrund der hohen Sensitivität der Zusammenhänge des Clusters C2 zur Trennung der verbleibenden Cluster mit mehreren Testdurchläufen zu evaluieren. In allen Testfällen werden die Anforderung L1 und L2 nicht erfüllt. Insgesamt wird in diesen C2-Testfällen die Anforderung DQ2 am stärksten verletzt, da im Vergleich zu den Testfällen Nr. 1 – 5 bzw. Nr. 6 - 10 die meisten Vorhersageänderungen zwischen den nicht-reduzierten Clustern stattfinden.

Weitere Testfälle mit veränderter Datenmenge werden nicht durchgeführt, da hierdurch keine weiteren Erkenntnisse hinsichtlich der Anforderungserfüllung von DQ2 erwartet werden. Die Verletzung der Anforderung durch das Gesamtmodell wird bereits durch die bestehenden Testfälle gezeigt. Durch eine Erhöhung der verbleibenden Daten in weiteren Testfällen ist es zwar möglich, den Grenzwert zu finden, ab dem die Anforderung DQ2 auch für die Cluster C0 und C2 erfüllt ist, jedoch ist dieses Wissen lediglich nützlich, um die minimale Datensatzgröße eines ähnlichen Problems festzulegen.

Fazit Anwendbarkeit

Die Anwendbarkeit der Anforderungen DQ1[294] und DQ2[295] wurde im Rahmen des Clustering-Algorithmus erfolgreich gezeigt. Die Anforderung DQ2 ist nur im Rahmen von Problemstellungen anwendbar, in denen unterschiedliche Klassen bzw. Cluster auftreten. Im Rahmen einer Regression existieren beispielsweise keine Klassen oder Gruppen, die im Einzelnen vermindert werden können.

Es wurden verschiedene Möglichkeiten zur Modellevaluation eines Clustering-Algorithmus vorgestellt und die resultierenden Ergebnisse diskutiert. Diese Evaluationsmöglichkeiten sind für Unsupervised-Lernansätze anwendbar. Die Problematik des relativen Vergleichs zweier Modellvorhersagen besteht darin, dass hierdurch keine sichere Aussage getroffen wird, welche der Vorhersagen korrekt ist und welche nicht. Daher ist die Aussagekraft dieser Evaluationsmethode eingeschränkt. Durch die Überprüfung der funktionalen Anforderungen wird diesem Nachteil teilweise begegnet. Im Fall eines Supervised-Learning-Problems ist es möglich, die Vorhersagen mit der Ground-Truth, die durch die

[294] Die gleichmäßige Änderung (bspw. über alle Klassen hinweg) der Datensatzgröße in den unterschiedlichen Entwicklungsphasen bis zu einem gewissen Schwellwert besitzt keine Auswirkungen auf die grundlegende Funktionalität des Modells.

[295] Alle beabsichtigten Klassen sind innerhalb des Trainingsdatensatzes für die grundlegende Funktionalität des Modells hinreichend vertreten. Die Veränderung der Klassenrepräsentanz einzelner Klassen verändert die Leistungsfähigkeit für jede andere beabsichtigte Klasse nicht.

Label des Datensatzes gegeben ist, abzugleichen und hierdurch die Bewertung der Leistungsfähigkeit durchzuführen.

Zur Auswahl der Daten, die nicht mehr zum Training des Algorithmus zur Verfügung stehen, sind je nach Problemstellung unterschiedliche Varianten möglich und sinnvoll. Im Rahmen von DQ1 wurden drei verschiedene Arten der Datenauswahl gezeigt, wobei alle Arten einen unterschiedlichen Einfluss auf das Modellergebnis besitzen. Aus diesen unterschiedlichen Ergebnissen lassen sich beispielsweise Anforderungen für weitere Datenerhebungen ableiten, durch die das Modell systematisch verbessert wird.

Dadurch, dass die Überprüfung der Anforderungen auf dem erneuten Training des Modells mit reduzierten Datensätzen beruht, hängt der zeitliche Aufwand zur Überprüfung der Anforderung stark von der Trainingszeit des zu überprüfenden Algorithmus ab.

Fazit Generalisierbarkeit

Durch die Untersuchung der Anforderungen DQ1 und DQ2 wird festgestellt, dass das originale Modell der Fahrstilzuordnung sensitiv auf die Änderung der Größe eines Clusters reagiert. Hierbei unterscheiden sich die einzelnen Cluster in ihrer Sensitivität. Das sportliche Cluster C2 beeinflusst die Leistungsfähigkeit der anderen, nicht-reduzierten Cluster am stärksten, was auf eine fehlende Generalisierbarkeit hinweist. Weitere Analysemöglichkeiten bilden an dieser Stelle die iterative Veränderung des originalen Modells hinsichtlich der verwendeten Eingangsgrößen, die erneute Überprüfung der funktionalen Anforderungen und die Untersuchung des Einflusses dieser Änderungen auf die Sensitivität des Modells, um im Rahmen der Anforderung DQ2 eine höhere Generalisierbarkeit zu erhalten.

Weitere Erkenntnisse über die Generalisierbarkeit des originalen Modells werden durch die Analyse von DQ1 bzw. der hierin verwendeten unterschiedlichen Methoden zur Auswahl der Datenuntermenge zum Training gewonnen. Die interindividuellen Unterschiede der Fahrer besitzen eine höhere Relevanz für die Unterscheidung der Fahrstile als die intraindividuellen, obwohl die unterschiedlichen Fahrten eines Fahrers ebenfalls zu verschiedenen Clustern zugeordnet werden. Es werden pro Fahrer maximal 30 Fahrten zum Training bei einer Anzahl von 32 unterschiedlichen Fahrern genutzt. Von den theoretisch möglichen 960 Fahrten stellen jedoch lediglich 404 Datenpunkte eine Manöverausführung des Linksabbiegens aus dem Stillstand dar, weshalb im Mittel ca. zwölf Datenpunkte pro Fahrer existieren. Die Analyse der Robustheit zeigt, dass ca. acht Fahrten pro Fahrer ausreichen, um eine annähernd gleiche Leistungsfähigkeit bzw. Generalisierbarkeit des Modells zu erhalten, wie durch die Nutzung der zwölf Datenpunkte.

Die Analyse der Anforderungen DQ1 und DQ2 sind geeignet, um die Generalisierbarkeit weiter zu untersuchen und Auswirkungen fehlender oder geeigneter Generalisierbarkeit festzustellen.

6.2.3.2 Datenvorverarbeitung DV1

- DV1: Mikroskopische Veränderungen der Vorverarbeitung der Eingangsdaten des Modells besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells.

Im Folgenden werden fünf Möglichkeiten vorgestellt, die Anforderung DV1 zu überprüfen. Drei der vorgestellten Methoden werden auf den Clustering-Algorithmus angewendet, um anschließend die generelle Anwendbarkeit der Anforderung DV1 sowie die gewonnenen Erkenntnisse hinsichtlich der Generalisierbarkeit des Anwendungsfalls zu diskutieren.

Zur Überprüfung dieser Anforderung besteht die erste Methode (DV1_M1) darin, die Rohdaten des originalen Trainingsdatensatzes in ihrer Aufbereitung, beispielsweise hinsichtlich der Filterung, zu verändern. Diese Daten werden dem unveränderten gelernten Modell (dem „originalen“ oder „finalen“ Modell) als „neue“ Eingangsdaten zur Verfügung gestellt. Die resultierende Vorhersage zur Clusterzuordnung der neuen Eingangsdaten, wird mit der des originalen, nicht veränderten, Trainingsdatensatzes hinsichtlich der Übereinstimmung verglichen. Eine hohe Übereinstimmung weist darauf hin, dass das originale Modell robust gegenüber mikroskopische Veränderungen der Datenverteilung reagiert und eine für diese Änderung ausreichende Generalisierbarkeit besitzt. Neben der reinen Übereinstimmung der Clusterzuordnung zwischen geänderten und originalen Datensatz wird die Einhaltung der funktionalen Anforderungen an das Modell durch diese geänderten Daten evaluiert, um bei Abweichungen zwischen der Clusterzuordnung der beiden Datensätze festzustellen, ob diese aus der unterschiedlichen Zuordnung von Randbereichen der funktionalen Anforderungen resultieren, oder ob die veränderte Vorverarbeitung funktional falsches Verhalten hervorruft. Der Nachteil der Methode DV1_M1 besteht darin, dass durch das Training des Modells mit den Daten, die in einer leicht veränderten Form zur Überprüfung herangezogen werden, die Ähnlichkeit zwischen Trainings- und Testdaten sehr groß ist. Es ist hierdurch möglich, dass Artefakte der originalen Datenpunkte erlernt werden, die durch eine veränderte Vorverarbeitung nicht berührt werden und hierdurch die hohe Übereinstimmung der Clusterzuordnung hervorgerufen wird. Dieses Ergebnis wird dann fälschlicherweise als ausreichende Generalisierbarkeit gewertet. Daher werden weitere Methoden diskutiert, um die Leistungsfähigkeit auch bei Verwendung „unbekannter“ Daten mit veränderter Vorverarbeitung zu überprüfen.

Zu diesem Zweck stellt eine Möglichkeit die Erhebung von Testdaten dar, die einer veränderten Vorverarbeitung unterzogen werden (Methode DV1_M2). Dabei ist darauf zu achten, dass diese Daten die gleichen Erhebungsbedingungen besitzen wie die des Trainingsdatensatzes, um die Auswirkungen dieser Daten auf die Veränderung der Vorverarbeitung und nicht auf andere Einflüsse zurückführen zu können. Für die Anwendung auf einen Unsupervised-Ansatz sind die „korrekten“ Ausgangsgrößen zunächst durch das originale Modell vorherzusagen, um diese anschließend mit denen der veränderten Datenpunkte zu vergleichen. Diese Methode besitzt den Nachteil eines hohen Aufwandes durch die erneute Datenerhebung.

Eine andere Möglichkeit (DV1_M3) besteht darin, dass die zur Überprüfung von DV1 benötigten Daten direkt vor dem Training des Modells aus dem Datensatz entfernt werden, um den Aufwand einer erneuten Datenerhebung zu vermeiden. Dieses auf dem reduzierten Datensatz trainierte Modell ist dann für die Systementwicklung und daher für die Überprüfung aller Anforderungen als das „finale“ bzw. „originale“ Modell zu nutzen. Die Testfall-Daten sind repräsentativ für den Gesamtdatensatz auszuwählen. Da im Rahmen des Entwicklungsprozesses von Supervised-Learning-Ansätzen ohnehin ein Testdatensatz extrahiert wird, ist es möglich, diesen für die Überprüfung der Anforderung DV1 zu modifizieren und zu nutzen. Da diese Testdatenextraktion vor dem Training des Algorithmus bei Entwicklungen aus dem Bereich des Unsupervised-Learning jedoch nicht vorgesehen ist, ist die Nutzung dieser Methode unwahrscheinlich, da die Extraktion der Testdaten bereits durch den Entwickler erfolgen müsste.

Eine Alternative für Unsupervised-Ansätze stellt daher die Nutzung der Ergebnisse der Anforderung DQ1 dar (Methode DV1_M4). In dieser Überprüfung wurden Modelle mit einem reduzierten Datensatz generiert, die eine hohe Übereinstimmung des funktionalen Verhaltens zum originalen Datensatz aufweisen. Die nicht zum Training verwendeten Daten stehen somit als Testdaten zur mikroskopischen Modifikation zur Verfügung. Da jedoch nur eine vergleichsweise geringe Datenmenge zum Test genutzt werden kann, wird diese Methode lediglich zusätzlich zu DV1_M1 genutzt, um deren Ergebnisse auf das Vorliegen von Artefakten hin zu untersuchen, die die hohe Clusterübereinstimmung verursachen. Ein ähnliches Vergleichsergebnis beider Methoden mit dem originalen Modell spricht dabei dafür, dass Artefakte nicht ursächlich für eine hohe Clusterübereinstimmung sind.

Neben diesen vier Methoden, deren Erkenntnisse alle auf der Modellvorhersage von modifizierten Testdaten beruhen, besteht eine fünfte Möglichkeit (DV1_M5) zur Überprüfung von DV1 in der Nutzung des analog zu DV1_M1 erzeugten veränderten Datensatzes als Trainingsdatensatz für ein neues Modell. Die Modellparameter sind dabei identisch zum originalen Modell zu halten. Die Vorhersagen des resultierenden Modells werden hinsichtlich ihrer Clusterübereinstimmung mit dem originalen Modell sowie der Einhaltung der funktionalen Anforderungen bzw. im Fall eines Supervised-Learning-Ansatzes mit der Ground-Truth überprüft. Es wird eine hohe Übereinstimmung in der Vorhersageleistung des veränderten Modells von Supervised Ansätzen erwartet, da die mikroskopische Veränderung des Datensatzes zu keiner Auf- oder Verdeckung relevanter Zusammenhänge führen sollte. Liegt diese hohe Übereinstimmung vor, so ist dies ein Zeichen einer hohen Übertragbarkeit der Modellstruktur, einer geeigneten Auswahl der Eingangsmerkmale und hierdurch einer ausreichenden Generalisierbarkeit im Rahmen der überprüften Veränderung. Das originale Modell sollte dabei natürlich entweder eine höhere Vorhersageleistung oder eine höhere Stabilität bzw. Generalisierbarkeit besitzen als das geänderte Modell. Da die Ground-Truth bei Unsupervised-Ansätzen nicht vorliegt, besteht lediglich die Möglichkeit, die Vorhersageleistungen des geglätteten Modells gegenüber dem originalen Modell zu vergleichen. Eine große Differenz der beiden Vorhersagen gilt es dabei näher zu

analysieren, da auch in diesen Ansätzen keine vorhersagerelevanten Veränderungen der Datenpunkte hervorgerufen werden sollten.

Im Folgenden werden die Methoden DV1_M1, DV1_M4 und DV1_M5 zur Überprüfung der Anforderung DV1 angewendet. Die Prinzipien der Implementierungen und Erkenntnisse der Methoden DV1_M2 und DV1_M3 sind analog zu denen der Kombination von DV1_M1 und DV1_M4, weshalb auf deren Überprüfung verzichtet wird. Zudem erfordert die Methode DV1_M2 die Erhebung eines neuen Datensatzes, was in der vorliegenden Arbeit nicht geleistet wird. Bevor die Implementierungen der Methoden diskutiert werden, wird die Datenstruktur sowie die mikroskopische Veränderung, die für alle Methoden Anwendung findet, vorgestellt.

Angewendete mikroskopische Datenveränderung

Das für die Datenerhebung des Anwendungsfalls genutzte Messsystem liest bereits vorverarbeiteten Sensorsignale aus dem Fahrzeug-CAN aus, weshalb eine weitere Verarbeitung, bspw. um ein potentielles Messrauschen zu entfernen, für das Training eines gelernten Modells nicht notwendig ist. Aus den Zeitreihen der Eingangsgrößen werden daher die statistischen Größen Minimum, Maximum und Standardabweichung direkt abgeleitet und zum Training des originalen Modells genutzt.[296] Dabei stellt die Verwendung dieser Merkmale selbst eine Filterung dar. Das Merkmal Ruck wird aus dem Signal der Längsbeschleunigung berechnet. Zur Überprüfung der Anforderung DV1 besteht eine Möglichkeit in der Anwendung von Glättungsmethoden, um einen mikroskopisch veränderten Datensatz zu erhalten. Es wird eine hohe Robustheit der erlernten Zusammenhänge auf diese Änderung erwartet, da durch eine Glättung die Amplituden der Extremwerte zwar verringert werden, jedoch über den gesamten Datensatz hinweg. Durch die Standardisierung der einzelnen Merkmale sollte die Glättung der Datenpunkte daher zu keiner gravierenden Änderung der Vorhersage führen. Wird dennoch eine hohe Sensitivität der Clustervorhersage auf diese mikroskopische Änderung in den verschiedenen Überprüfungsmethoden festgestellt, ist daher die Generalisierbarkeit des originalen Modells in Frage zu stellen. Neben der Glättung sind auch andere Arten der Veränderung der Eingangsdaten möglich, wie beispielsweise der Einsatz eines Tiefpassfilters. Eine Filterung ist im vorliegenden Anwendungsfall aufgrund der vier unterschiedlichen Sensorquellen für die verschiedenen Eingangssignale für jede Sensorquelle individuell zu parametrisieren, wodurch das Risiko einer ungünstigen bzw. inkorrekten Parametrierung besteht, welches die inhärenten Zusammenhänge in den Daten verändert. Hierdurch ist es möglich, dass die durch die Filterung resultierende Änderung der Leistungsfähigkeit der Vorhersage des originalen Modells (DV1_M1), des reduzierten Modells (DV1_M4) oder gelernten Modells (DV1_M5) nicht alleine auf dessen geringe Robustheit zurückzuführen ist. Für die Überprüfung der An-

[296] Eine detaillierte Beschreibung der Prozesskette von den CAN-Signalen zu den extrahierten Merkmalen ist in A.4 gegeben.

wendbarkeit der Anforderung DV1 wird daher eine Glättungsfunktion verwendet, die über alle Eingangsgrößen hinweg die gleiche Parametrierung besitzt.

Zur Glättung wird unter anderem der gleitende Mittelwert (MA)[297] angewendet. Die berechneten Mittelwerte ergeben dabei die neuen, geglätteten Datenpunkte. Die Randbereiche des geglätteten Signals werden hierdurch jedoch entfernt.[298] Die Größe der Untermenge an Datenpunkten, die gemittelt werden, wird auch als „Glättungsfenster" bezeichnet und bestimmt die Stärke der Glättung. Diese Glättung resultiert, wie bereits beschrieben, in geringeren Extremwerten.

Neben dem einfachen gleitenden Mittelwerts werden ebenfalls Testfälle mit einer Spezialform des gleitenden Mittelwerts, dem triangularen gleitenden Mittelwerts (TMA)[299], angewendet. Die Spezialform stellt einen linear gewichteten gleitenden Mittelwert dar, wobei die Gewichtung innerhalb des Glättungsfensters in Dreiecksform verläuft. Er wird im ersten Schritt analog zum einfachen gleitenden Mittelwert berechnet. Im zweiten Schritt werden die hieraus resultierenden arithmetischen Mittelwerte nochmals mit dem gleichen[300] Glättungsfenster gemittelt. Dies ist analog zu einer einmaligen Glättung mit einem doppelt so breiten, quadratisch gewichteten Fenster. Hieraus entsteht ein stärker geglätteter Verlauf der Zeitreihe als bei Verwendung eines MA.[301]

Für die Testfalldurchführung der Anforderung in den verschiedenen Methoden wird einerseits die Parametrierung des Glättungsfensters verändert und andererseits das Glättungsverfahren, um möglichst unterschiedliche Veränderungsarten zu überprüfen.

Methode DV1_M1

In Methode DV1_M1 werden die Clusterzuordnungen des geglätteten Gesamtdatensatz als „neue" Eingangsgrößen durch das originale Modell vorhergesagt. Die Höhe der Übereinstimmung der Vorhersage des unveränderten Datenpunkts im Vergleich zum geglätteten Datenpunkt sagt aus, welche Robustheit das originale Modell hinsichtlich einer veränderten Vorverarbeitung, wie sie im Betrieb möglich ist, besitzt. In Tabelle 6-3 sind die Ergebnisse der Testfälle zusammengefasst.

---

[297] Englisch: Moving Average.

[298] Vgl. Moving Average (2008).

[299] Englisch: Triangular Moving Average.

[300] „gleich" aufgrund der genutzten Matlab-Implementierung. Vgl. MATLAB: Financial Toolbox Documentation. Moving average – MATLAB tsmovavg (2019).

[301] Vgl. Kirkpatrick, C.; Dahlquist, J.: Technical Analysis (2010), S. 285.

Tabelle 6-3: Testfälle der Anforderung DV1 (Methode DV1_M1)

Nr.	**Übereinstimmung aller Cluster**	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Veränderung**
1	91 %	ja	ja	ja	MA (Glättungsfenster 2)
2	90 %	ja	ja	ja	MA (Glättungsfenster 3)
3	91 %	ja	ja	ja	MA (Glättungsfenster 4)
4	91 %	ja	ja	ja	MA (Glättungsfenster 5)
5	90 %	ja	ja	ja	MA (Glättungsfenster 10)
6	90 %	ja	ja	ja	MA (Glättungsfenster 20)
7	90 %	ja	ja	ja	TMA (Glättungsfenster 2)
8	90 %	ja	ja	ja	TMA (Glättungsfenster 3)
9	90 %	ja	ja	ja	TMA (Glättungsfenster 4)
10	90 %	ja	ja	ja	TMA (Glättungsfenster 5)
11	90 %	ja	ja	ja	TMA (Glättungsfenster 10)
12	89 %	ja	ja	ja	TMA (Glättungsfenster 20)

Im ersten Testfall wird ein Glättungsfenster von zwei Datenpunkten bei der Anwendung des gleitenden Mittelwerts MA (2) verwendet. Zwei Datenpunkte besitzen hierbei einen Abstand von ca. 20 ms[302], wodurch sich das Glättungsfenster zu 40 ms ergibt. Um die Höhe der Zuordnungs-Übereinstimmung von 91% hinsichtlich der Robustheit des Modells einzuordnen, werden die Datenpunkte, die nicht übereinstimmen, näher analysiert. Die Erfüllung der funktionalen Anforderung weist darauf hin, dass die veränderte Vorhersage dieser Datenpunkte durch deren Lage an den Grenzen der Clusterbereiche hervorgerufen wird. Zur Überprüfung dieser Vermutung werden die originalen, nicht-geglätteten Datenpunkte durch eine Hauptkomponentenanalyse[303] (PCA)[304] niedrigdimensional visualisiert und entsprechend ihrer originalen Clusterzuordnung markiert. Diese Darstellung ist in Abbildung 6-19 (links) gegeben. Liegen die nicht-übereinstimmenden Datenpunkte in ihrer ursprünglichen, nicht-geglätteten Form an den Clustergrenzen, so ist die veränderte Zuordnung aufgrund der mikroskopischen Veränderung hierdurch erklärbar. Wie in Abbildung 6-19 in der rechten Darstellung durch die schwarzen Kreuze markiert, ist dies der Fall,

[302] Der Zusatz "ca." resultiert aus Schwankungen des zeitlichen Abstandes von zwei Datenpunkten, die sich beispielsweise durch leicht schwankende Berechnungsdauern des Messsystems ergeben, da die Daten in Echtzeit aufgenommen und verarbeitet werden.

[303] Vgl. Bishop, C. M.: Pattern recognition and machine learning (2006), S. 561ff.

[304] Englisch: Principal Component Analysis

weshalb die Übereinstimmung der Clusterzuordnungen von 91 % als Nachweis ausreichender Generalisierbarkeit des originalen Modells gewertet wird. Lediglich zwei der nicht-übereinstimmenden Datenpunkte besitzen einen höheren Abstand zu den Clustergrenzen, der nicht alleine auf eine mikroskopische Änderung zurückzuführen ist.

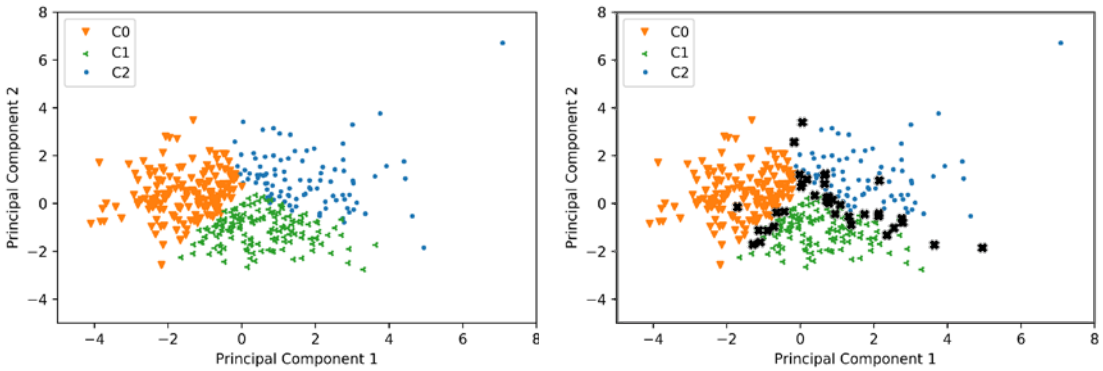


Abbildung 6-19: Lage der nicht-übereinstimmenden Datenpunkte (Testfall Nr. 1)

Eine Verbreiterung des Glättungsfensters in den Testfällen Nr. 2 - 6 führt zu vergleichbaren Übereinstimmungswerten der Clusterzuordnung der originalen zu veränderten Datenpunkten, wobei eine sinkende Tendenz der Übereinstimmungswerte bei der Verwendung einer geringeren Rundung erkennbar ist. Dies erklärt sich durch die immer stärkere mikroskopische Veränderung der Werte. Der Testfall Nr. 2 stellt dabei einen Ausreißer dieser Tendenz dar, der jedoch auf eine ungünstige veränderte Clusterzuordnung in dessen Grenzbereichen zurückführbar ist. Auch eine Änderung des Glättungsverfahrens in den Testfällen Nr. 7 – 12 birgt die gleiche grundsätzliche Erkenntnis dieser Methode der Anforderungsüberprüfung DV1: Die Generalisierbarkeit des originalen Modells ist zur Vorhersage von Daten mit einer anderen Vorverarbeitung, die in einer mikroskopischen Änderung der Merkmalswerte resultiert, ausreichend hoch. Allerdings gilt hierbei die bereits erwähnte Einschränkung dieser Aussage, dass die scheinbare Generalisierbarkeit auch durch gelernte Artefakte der Datenpunkte hervorgerufen werden kann, da die gleichen Datenpunkte zum Training und in leicht veränderter Weise zum Test genutzt werden.

Methode DV1_M4

Die Methode DV1_M4 beruht auf der Nutzung eines bereits reduzierten Modells, welches vergleichbare Eigenschaften zum originalen Modell besitzt. Die nicht zum Training genutzten Daten stehen für mikroskopische Veränderungen zur Verfügung. Da die für den Test zur Verfügung stehende Datenmenge je nach Robustheit des originalen Modells auf Reduzierung der Daten klein ausfällt, wird diese Methode lediglich als Ergänzung der Überprüfungsmöglichkeit DV1_M1 genutzt, um die hieraus resultierenden Erkenntnisse zu bestätigen oder widerlegen. Für den Anwendungsfall wird das reduzierte Modell des Testfalls Nr. 9 der Anforderung DQ1 (siehe Tabelle 6-1) genutzt, da eine Übereinstimmung der Clusterzuordnung von 97 % zwischen diesem Modell und dem Original vorherrscht. Wird festgestellt, dass die Werte der Clusterübereinstimmung in den korrespondierenden Testfällen zwischen DV1_M1 und DV1_M4 ähnlich ausfallen, gilt die Erkenntnis der Methode

DV1_M1 als bestätigt. Die Ergebnisse der Testfälle der Methode DV1_M4 sind in Tabelle 6-4 aufgeführt.

Tabelle 6-4: Testfälle der Anforderung DV1 (Methode DV1_M4)

Nr.	**Übereinstimmung aller Cluster**	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Veränderung**
1	87 %	ja	diskutabel	ja	MA (Glättungsfenster 2)
2	86 %	ja	diskutabel	ja	MA (Glättungsfenster 3)
3	86 %	ja	diskutabel	ja	MA (Glättungsfenster 4)
4	86 %	ja	diskutabel	ja	MA (Glättungsfenster 5)
5	86 %	ja	diskutabel	ja	MA (Glättungsfenster 10)
6	87 %	ja	diskutabel	ja	MA (Glättungsfenster 20)
7	91 %	ja	diskutabel	ja	TMA (Glättungsfenster 2)
8	91 %	ja	diskutabel	ja	TMA (Glättungsfenster 3)
9	90 %	ja	diskutabel	ja	TMA (Glättungsfenster 4)
10	90 %	ja	diskutabel	ja	TMA (Glättungsfenster 5)
11	89 %	ja	diskutabel	ja	TMA (Glättungsfenster 10)
12	88 %	ja	diskutabel	ja	TMA (Glättungsfenster 20)

Die Übereinstimmung der Clustervorhersage in den Testfällen, in denen der triangulare gleitende Mittelwert angewendet wird (Nr. 7 – Nr. 12), ist sehr ähnlich zu den korrespondierenden Testfällen der Methode DV1_M1. Dies scheint die Erkenntnisse aus DV1_M1 zu bestätigen. Jedoch wird im Rahmen der Anwendung des einfachen Mittelwerts eine um ca. 4 % geminderte Übereinstimmung erreicht, wodurch die nicht-übereinstimmenden Datenpunkte einer detaillierteren Analyse unterzogen werden. Wie bereits in DV1_M1 angewendet, wird die Lage der nicht-übereinstimmenden Datenpunkte in der Darstellung der Hauptkomponentenanalyse des originalen Datensatzes mit der originalen Clusterzuordnung untersucht. Die betreffenden Datenpunkte befinden sich bis auf einen in den Grenzbereichen der Cluster, wodurch die Differenzen der Clusterzuordnung erklärbar sind. Der Datenpunkt, der sich nicht direkt im Grenzbereich befinden, ist ebenfalls in DV1_M1 falsch zugeordnet, was die Übereinstimmung der Aussagen dieser Methoden unterstreicht.[305] Die Differenz von 4 % entspricht drei Datenpunkten des Testdatensatzes, die in DV1_M4 inkorrekt zugeordnet werden. Bedingt durch die zufällige Auswahl an Testdatenpunkten im Rahmen des Testfalls Nr. 9 der Anforderung DQ1 ist es möglich, dass die

[305] Siehe Anhang 8-15.

relative Menge an Datenpunkten innerhalb der Clustergrenzbereiche in den Testfällen der Methode DV1_M4 gegenüber denen der in DV1_M1 erhöht ist.

Die Bewertung der Anforderungserfüllung L2 mit „diskutabel“ in allen Testfällen beruht auf der nicht möglichen Trennbarkeit der Cluster C0 und C1 entsprechend der Anforderung. Dies ist ebenfalls auf die Auswahl an Testdaten zurückführbar, da die für die Trennung notwendigen Datenpunkte in den Grenzbereichen dieser Anforderung nicht im Testdatensatz vorhanden sind.[306] Die Erkenntnisse der Methode DV1_M1 bezüglich der Generalisierbarkeit des originalen Modells werden daher trotz zunächst widersprüchlich erscheinenden Ergebnissen durch die Methode DV1_M4 bestätigt.

Methode DV1_M5

Der geglättete Gesamtdatensatz wird im Rahmen der Methode DV1_M5 zum Training eines neuen Modells verwendet. Die hieraus resultierenden Clusterzuordnungen werden mit denen des unveränderten Datensatzes aus dem originalen Modell verglichen. Durch diesen Vergleich werden Erkenntnisse hinsichtlich der Generalisierbarkeit der Grundstruktur bzw. der ausgewählten Merkmale des originalen Modells gewonnen. Eine hohe Übereinstimmung der Clusterzuordnung spricht beispielsweise dafür, dass die Auswahl der einzelnen Merkmale geeignet ist, um sie auf ähnliche Problemstellungen zu übertragen. Die Ergebnisse der Testfälle sind in Tabelle 6-5 zusammengefasst.

[306] Siehe Anhang 8-16.

Tabelle 6-5: Testfälle der Anforderung DV1 (Methode DV1_M5)

Nr.	**Übereinstimmung aller Cluster**	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Veränderung**
1	83 %	ja	ja	ja	MA (Glättungsfenster 2)
2	72 %	ja	ja	nein	MA (Glättungsfenster 3)
3	72 %	ja	ja	nein	MA (Glättungsfenster 4)
4	72 %	ja	ja	nein	MA (Glättungsfenster 5)
5	71%	ja	ja	nein	MA (Glättungsfenster 10)
6	71 %	ja	ja	nein	MA (Glättungsfenster 20)
7	84 %	ja	ja	ja	TMA (Glättungsfenster 2)
8	83%	ja	ja	diskutabel	TMA (Glättungsfenster 3)
9	82 %	ja	ja	diskutabel	TMA (Glättungsfenster 4)
10	82 %	ja	ja	diskutabel	TMA (Glättungsfenster 5)
11	81 %	ja	ja	diskutabel	TMA (Glättungsfenster 10)
12	80 %	ja	ja	diskutabel	TMA (Glättungsfenster 20)

Die funktionalen Anforderungen des ersten Testfalls werden erwartungsgemäß erfüllt, doch die Clusterübereinstimmung von 83 % zwischen mit dem geglätteten Datensatz gelernten und dem originalen Modell fällt für diese schwache Glättung der Eingangssignale gering aus. Eine Analyse der aus den Signalen berechneten Merkmale zeigt, dass die Merkmale des maximalen und minimalen Rucks eine große Abweichung zwischen den geglätteten und den originalen Werten besitzen. Diese resultiert daraus, dass der Ruck als Ableitung der Längsbeschleunigung berechnet wird und diese Beschleunigung im genutzten CAN-Signal in diskreten Stufen vorliegt. Durch die Glättung wird die Steigung zwischen den Stufen verändert, was in einem veränderten Ruck resultiert.[307] Es wird durch die Glättung der Längsbeschleunigung über mehrere Datenpunkte hinweg teilweise eine höhere Steigung des geglätteten Verlaufs erzielt, als zwischen zwei Datenpunkten vorliegt, die zur Berechnung des Rucks ohne Glättung herangezogen werden. Hierdurch ändert sich die Lage des Maximums des Rucks, wodurch anstelle einer mikroskopischen Änderung eine makroskopische Änderung des Merkmals Ruck erreicht wird.[308] Daher wird durch die Glättung im Gegensatz zu den anderen Merkmalen kein linearer Zusammenhang der geänderten Ruckmerkmale gegenüber den originalen Merkmalen erreicht. Dies zeigt Abbildung

[307] Der Vergleich zwischen dem Längsbeschleunigungssignal mit und ohne Glättung ist exemplarisch in Anhang 8-17 gegeben.

[308] Ein exemplarischer Verlauf mit veränderter Lage der Maximalwerte zwischen geglättetem und originalen Verlauf der Längsbeschleunigung ist in Anhang 8-18 dargestellt.

6-20 mit dem Vergleich der Merkmalsänderungen zwischen geglätteten und originalem Merkmalswert von maximaler Längsbeschleunigung zu maximalem Ruck.

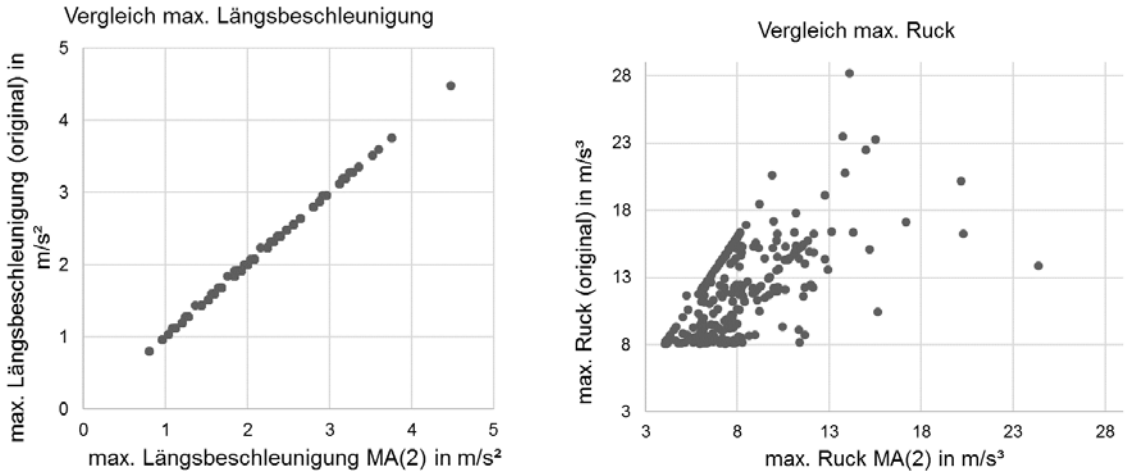


Abbildung 6-20: Zusammenhang zwischen geglätteten und originalen Merkmalen

Hierdurch erklärt sich die Differenz der Clusterzuordnungen bzw. der inhärent enthaltenen Zusammenhänge der originalen Modellvariante gegenüber der des Testfalls Nr. 1 trotz Standardisierung der Merkmale des Rucks. Auch die maximalen Ruckwerte des Testfalls Nr. 2, welcher ein geringfügig breiteres Glättungsfenster im Vergleich zu Testfall Nr. 1 besitzt, zeigen diese makroskopische Änderung aufgrund der weiteren Lageveränderung. Die Veränderung der maximalen Ruckwerte zwischen dem originalen Datensatz und dem Datensatz aus Testfall Nr. 2 wird in Abbildung 6-21 links dargestellt. Dass auch der Zusammenhang zwischen der max. Ruckwerte aus Testfall Nr. 1 zu Nr. 2 eine makroskopische Änderung erhalten wird durch die rechte Darstellung verdeutlicht.

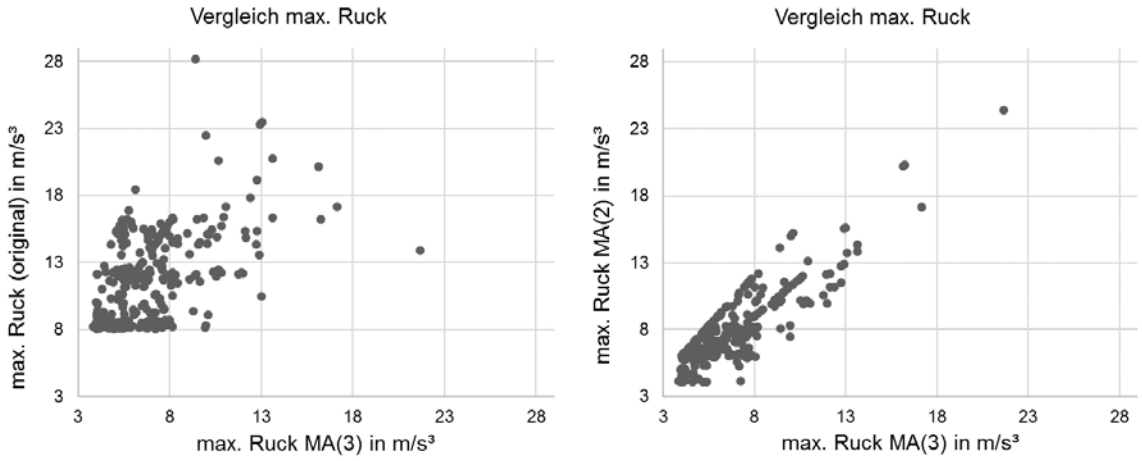


Abbildung 6-21: Zusammenhang zwischen max. Ruckwerten (Testfall Nr. 1 und 2)

Hieraus resultiert die weitere Verminderung der Clusterübereinstimmung zum originalen Modell. Die Anforderung L3 wird durch das aus Testfall Nr. 2 resultierende Modell nicht erfüllt, da die Ordinalität der Cluster C1 und C2 bzw. „ausgeglichen“ und „sportlich“ vertauscht ist. Eine Analyse der Veränderungen in der Clusterzuordnung der Datenpunkte

zwischen Testfall Nr. 1 und Testfall Nr. 2 zeigt, dass von 57 veränderten Zuordnungen knapp 50 % daraus resultieren, dass Cluster 0 und 1 (aus Testfall Nr. 1) zu Cluster 2 (Testfall Nr. 2) zugeordnet werden. Die Merkmalswerte dieser Datenpunkte unterscheiden sich hierbei fast ausschließlich in den Merkmalen des Rucks. Hierdurch lässt sich der Anstieg des Verlaufs der kumulierten Häufigkeit der Lenkradwinkelgeschwindigkeit im Vergleich zu Testfall Nr. 1 erklären. Die weiteren Testfälle Nr. 3 – 6 unterscheiden sich in der Größe des Glättungsfensters voneinander. Die Übereinstimmung in der Clusterzuordnung sinkt mit steigender Fenstergröße geringfügig. Dieser in Relation zu Nr. 1 und Nr. 2 gesehene geringe Abfall der Clusterübereinstimmung resultiert daraus, dass die Lage der Extremwerte des Merkmals Rucks ab einer Fenstergröße von vier stabil sind. Hierdurch werden die Merkmalswerte lediglich mikroskopisch verändert, was zu ähnlicheren Modellvarianten ab dieser Fenstergröße führt.[309]

Wie bereits in Testfall Nr. 1 aufgetreten, sinkt auch bei der Verwendung des triangularen gleitenden Mittelwerts bei kleinem Glättungsfenster in Testfall Nr. 7 die Clusterübereinstimmung stark im Vergleich zu den Erwartungen. Analog zu Testfall Nr. 1 ist dieser Abfall durch auf Glättung der Längsbeschleunigung und die hieraus resultierende Veränderung der Lagepunkte der Extremwerte des Rucks zurückzuführen. Durch die doppelte Glättung des triangularen gleitenden Mittelwerts im Vergleich zum einfachen Mittelwert ändert sich die Steigung des Längsbeschleunigungsverlaufs bei wachsendem Glättungsfenster in den nachfolgenden Testfällen geringfügig. Dies führt dazu, dass zwischen Testfall Nr. 7 und Nr. 8 die Anzahl der Datenpunkte mit einer makroskopischen Änderung geringer ist, als im Vergleich des Übergangs zwischen Testfall Nr. 1 und 2 (siehe Abbildung 6-21, rechts). Abbildung 6-22 (links) zeigt diesen Übergang zwischen Testfall Nr. 7 und 8. Hierdurch ist der Abfall der Clusterübereinstimmung zwischen diesen Testfällen relativ gesehen geringer. Ab einem Glättungsfenster von vier (Testfall Nr. 9) besteht, analog zur Glättung mit dem einfachen triangularen Mittelwert, ein linearer Zusammenhang zwischen den Glättungsstufen (siehe Abbildung 6-22, rechts), woraus ein lediglich schwacher Abfall der Clusterübereinstimmung ab dieser Fenstergröße in den weiteren Testfällen resultiert. Dass der Abfall der Clusterübereinstimmung zwischen Testfall Nr. 7 und 8 aufgrund der Angaben aus Tabelle 6-5 genauso hoch zu sein scheint, wie zwischen Nr. 8 und 9 liegt in der Rundung der Übereinstimmungswerte begründet. Die tatsächliche Differenz zwischen diesen beiden Testfallübergängen beträgt 1,3 % (Nr. 7 auf Nr. 8) und 0,3 % (Nr. 8 auf Nr. 9).

[309] Der Vergleich zwischen dem Merkmal des maximalen Rucks von Testfall Nr. 2 und Nr. 3 ist in Anhang 8-19 dargestellt.

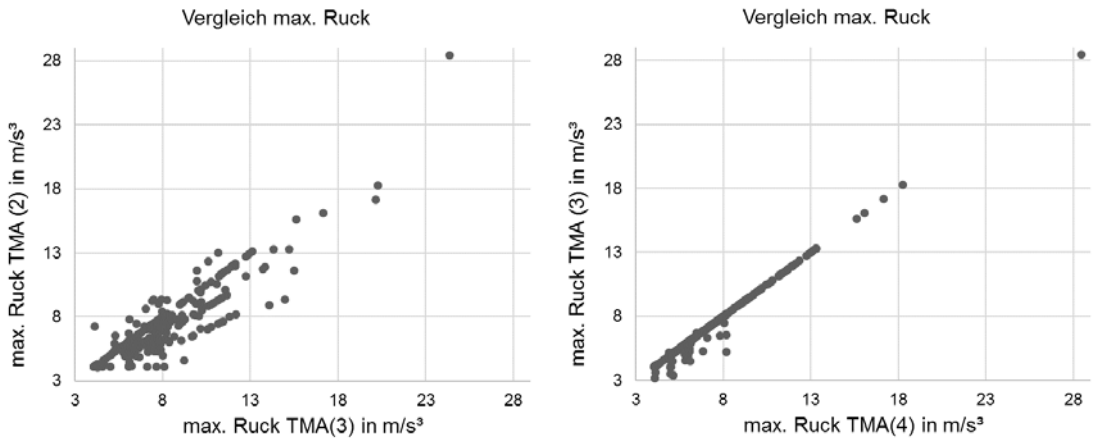


Abbildung 6-22: Zusammenhang zwischen max. Ruckwerten (Testfall Nr. 7, 8 und 9)

Die funktionalen Anforderungen werden im Testfall Nr. 7 komplett erfüllt, was sich jedoch mit Testfall Nr. 8 ändert. Die kumulierten Häufigkeitsverläufe der Lenkradwinkelgeschwindigkeit der Cluster C1 und C2 sind ab diesem Testfall zur Erfüllung der Anforderung L3 nichtmehr klar trennbar.[310]

Die verwendete Glättung führt zu einer makroskopischen Änderung der Datenpunkte von einzelnen Merkmalswerten. Hierdurch wird nicht die Anforderung DV1 überprüft, welches die ursprüngliche Intention dieser Testfälle ist, sondern die Abbildung der Realität durch die Eingangsmerkmale. Durch die diskreten Stufen der Längsbeschleunigung des originalen Datensatzes berechnen sich die Extremwerte des Rucks teilweise nicht der Realität entsprechend. Allerdings sind auch diese, teilweise nicht der Realität entsprechenden Ruckwerte zur Trennung der einzelnen Fahrstile bzw. Cluster relevant, was im Rahmen der Überprüfung der funktionalen Anforderungen (Abschnitt 6.2.2) festgestellt wird. Die Anwendung der Methode DV1_M5 wirft jedoch die Frage auf, ob die Verwendung eines geglätteten oder anderweitig vorverarbeiteten Datensatzes für das finale Modell besser geeignet wäre. Denn auch die Clusterzuordnung des bisherigen finalen Modells stellt keine Ground-Truth dar. Der relative Vergleich der Clusterzuordnungen zwischen geglätteten und nicht-veränderten Modell hinsichtlich der Clusterübereinstimmung besitzt daher lediglich die Aussage, dass Differenzen der beiden Modelle vorliegen, nicht welches der Modelle die höhere Leistungsfähigkeit besitzt. Beispielsweise erfüllen die Modelle der Testfälle Nr. 1 und Nr. 7 alle funktionalen Anforderungen. In der Anforderung L3 wird sogar in den geglätteten Modellen eine höhere Trennbarkeit der Cluster C1 und C2 erzielt als im nicht geglätteten Modell.[311] Der Vergleich der Lückenakzeptanzkurven, welche zur Evaluation des Modellergebnisses während der Entwicklung genutzt wurden (siehe Abschnitt 6.1.4), zeigt in allen drei Fällen (nicht geglättet, MA (2) geglättet und TMA (2) geglättet) einen plausiblen Verlauf, wie in Abbildung 6-23 dargestellt.

[310] Siehe Anhang 8-20.

[311] Siehe Anhang 8-21.

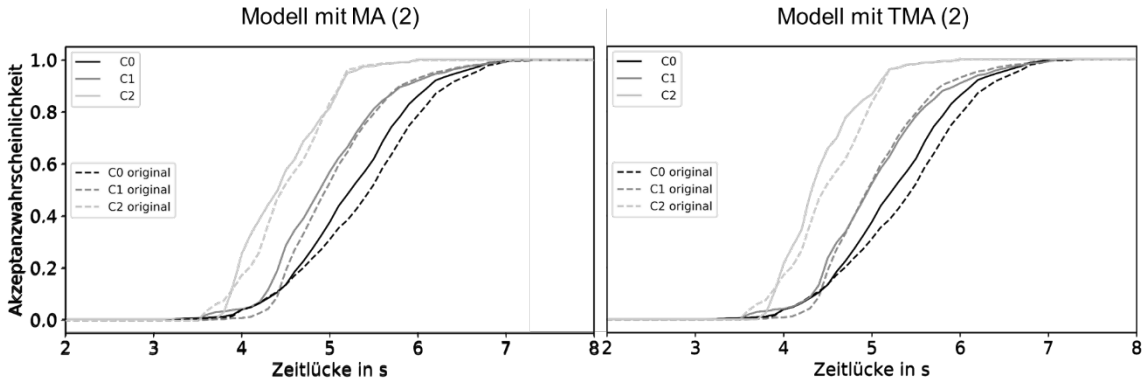


Abbildung 6-23: Vergleich der Akzeptanzkurven der Modelle

In den geglätteten Modellen fällt die Überschneidung der Kurven des Cluster C1 mit C0 deutlich geringer aus, dafür sind die Abstände zwischen den Kurven C1 und C0 nicht so stark ausgeprägt wie im originalen Modell. Eine objektive Bewertung der relativen Leistungsfähigkeit dieser Modelle ist ohne weitere Untersuchungen nicht möglich, da der rein visuelle Vergleich keine offensichtlich höhere oder niedrigere Leistungsfähigkeit offenbart. Auch eine Aussage zur Erfüllung der anderen Robustheitsanforderungen, um die Generalisierbarkeit der geglätteten Modelle zu überprüfen, kann ohne weitere Untersuchungen nicht getroffen werden. Zudem ist zu analysieren, ob mit dem geglätteten Datensatz eine andere Eingangsmerkmalskonfiguration ein besseres Modellergebnis erreicht, da die bisherige Auswahl der Eingangsgrößen auf die nicht geglättete Berechnung des Rucks ausgelegt wurde. Diese Untersuchungen übersteigen jedoch den Rahmen der vorliegenden Arbeit, da das Ziel in der Überprüfung der Anwendbarkeit des vorgestellten Ansatzes zur Überprüfung fehlender Generalisierbarkeit besteht, welches durch die bisherigen Untersuchungen und Erkenntnisse bereits erfüllt ist.

Fazit Anwendbarkeit

Die Anwendbarkeit der Anforderung DV1[312] wurde erfolgreich unter Nutzung von Glättungsverfahren durch drei verschiedene Methoden gezeigt. Andere Veränderungsverfahren der Daten zur Generierung einer mikroskopischen Modifikation sind ebenfalls möglich. Es ist darauf zu achten, dass diese Veränderung den wahren Informationsgehalt der Daten nicht verfälschen. Auch bei Vorliegen von Bilddaten ist eine mikroskopische Veränderung beispielsweise durch Filterung möglich. Die angewendeten Methoden zur Überprüfung stellen jedoch nicht den Fall dar, mit dem das originale Modell durch eine mikroskopische Veränderung der Datenpunkte bestmöglich auf Robustheit gestört wird, sondern sind Repräsentanten einer im Systemlebenszyklus möglichen Veränderung. Diese Wahl begründet sich mit der expliziten Überprüfung des Vorliegens einer Sensitivität auf eine bestmögliche Störung durch mikroskopische Veränderung[313] im Rahmen der direkten Ursachenüberprü-

[312] Mikroskopische Veränderungen der Vorverarbeitung der Eingangsdaten des Modells besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells.

[313] Siehe E24: Sensitivität auf Störungen (adverserial examples) in Abschnitt 4.1.2.

fung von fehlender Generalisierbarkeit, wie in Unterkapitel 5.3 vorgestellt. Durch die Überprüfung auf Sensitivität auf eine realitätsnahe anstelle einer optimalen Störung wird bei Nichtbestehen dieser Anforderung direkt notwendiges Handlungspotential aufgedeckt. Dieses Handlungspotential besteht entweder in Form der Verbesserung des Modells hinsichtlich der Generalisierbarkeit oder in Form von Maßnahmen, die die überprüfte Veränderung im Betrieb des Modells ausschließen.

Es existieren zwei weitere Methoden (DV1_M2 und DV1_M3), um die Anforderung DV1 zu überprüfen, wobei die Methode DV1_M3 vor allem für Supervised-Learning-Ansätze geeignet ist. Die Anwendung und Evaluierung der Methoden sind analog zu DV1_M4.

Die Evaluierung der Modellergebnisse fand für den vorliegenden Fall eines Unsupervised-Ansatzes einerseits anhand der Übereinstimmung zwischen der Eingruppierung der Datenpunkte durch das originale zum veränderten Modell und andererseits anhand der Überprüfung der funktionalen Anforderungen statt. In der Anwendung DV1_M4 wird deutlich, dass die Ergebnisse dieser Methode kritisch hinsichtlich ihrer quantitativen Aussagekraft zu hinterfragen sind, da sie auf einer geringen Anzahl an Testdaten basieren. Dies gilt ebenfalls für die Methode DV1_M2 und DV1_M3, da der hierin verwendete Testdatensatz eine geringe Größe besitzt. Für einen Supervised-Ansatz besteht neben der Möglichkeit des relativen Vergleichs zwischen originaler und veränderter Modellvorhersage auch die des direkten Vergleichs zur Ground-Truth eines Testdatensatzes, welcher nicht zum Training, sondern nur zur Evaluierung genutzt wird. Hierdurch sind exaktere Aussagen zur Leistungsfähigkeit bzw. Generalisierbarkeit des Modells möglich.

Fazit Generalisierbarkeit

Die Methoden DV1_M1 bis DV1_M4 bergen Erkenntnisse hinsichtlich der Zuverlässigkeit der Vorhersage des originalen Modells, wenn im Betrieb Eingangsdaten mit einer veränderten Vorverarbeitung auftreten. Hierdurch wird die Generalisierbarkeit des originalen Modells auf unbekannte Daten, die eine mikroskopisch veränderte Struktur besitzen, überprüft. Berufen sich die Vorhersagen des originalen Modells beispielsweise auf Artefakte innerhalb der Trainingsdaten, wird dies im Rahmen dieser Überprüfung aufgedeckt.

Die Methode DV1_M5 untersucht die Stabilität der Modellstruktur bei der Verwendung eines mikroskopisch modifizierten Datensatzes. Hierdurch wird beispielsweise überprüft, ob das Modell, welches auf einem veränderten Datensatz trainiert wurde, eine höhere Leistungsfähigkeit besitzt. Wenn die Vorverarbeitung des originalen Modells die enthaltenen Informationen und Zusammenhänge bestmöglich hervorhebt, hat dieses veränderte Modell eine geringere Vorhersageleistung zu besitzen.

Im vorliegenden Anwendungsfall wird eine ausreichende Generalisierung des originalen Modells durch die Methoden DV1_M1 und DV1_M4 nachgewiesen. In Verbindung mit den Ergebnissen der Methode DV1_M5 ist das jedoch unerwartet, da sich die Merkmale des Rucks nicht nur mikroskopisch, sondern teilweise auch makroskopisch verändern. Das wiederum hebt hervor, dass das originale Modell auch bei teilweise makroskopisch gestörten Ruckmerkmalen eine gute Vorhersageleistung besitzt, was auf eine Generalisierbarkeit

der Zusammenhänge der anderen Eingangsmerkmale schließen lässt. Allerdings wird durch die Methode DV1_M5 aufgedeckt, dass die Berechnung des Merkmals Rucks nicht der Intention in der Entwicklung im originalen Modell entspricht. Hierdurch sind weitere Analysen sinnvoll, ob ein besseres Vorhersageergebnis beispielsweise hinsichtlich der Trennbarkeit der Clusterverläufe der Lückenakzeptanzkurven durch eine veränderte Datenvorverarbeitung und einer veränderten Auswahl an Merkmalsgrößen erreicht wird.

Die Analyse der Anforderungen DV1 ist geeignet, um die Generalisierbarkeit weiter zu untersuchen und Auswirkungen fehlender oder geeigneter Generalisierbarkeit festzustellen.

6.2.3.3 Abdeckung A1

- A1: Die Funktion des Modells ist auch in dessen Bereichen mit einer spärlichen Abdeckungsrate durch die Trainingsdaten gewährleistet.

Zur Überprüfung der Anforderung ist es notwendig, gezielt Testdaten für die Bereiche mit einer geringen Abdeckung zu generieren. Hierzu wurden drei unterschiedliche Methoden identifiziert, die sich hinsichtlich ihres Aufwands und Anwendbarkeit im normalen Entwicklungsprozess unterscheiden.

Die erste Methode (A1_M1) besteht darin, die Testdaten durch eine erneute Datenaufnahme zu erheben. Die Notwendigkeit von Testdaten, die bereits im Trainingsdatensatz kaum oder nicht vorhanden sind, erfordert entweder die Datenaufnahme in expliziten Testfällen, durch die diese gering auftretenden Daten zuverlässig und dem späteren Betrieb entsprechend erhoben werden, oder die Aufnahme einer hohen Anzahl an erneuten Testdaten, damit die Wahrscheinlichkeit hoch genug ist, dass die benötigten Daten enthalten sind. Im vorliegenden Anwendungsfall tritt durch die Aufnahme von neuen, unbekannten Daten das Problem auf, dass keine zuverlässigen Label dieser Daten vorliegen und daher die Bewertung des funktional korrekten Verhaltens nur durch die in der Literatur identifizierten Zusammenhänge zu evaluieren ist. Durch den hohen Aufwand, der mit dieser Methode verbunden ist, in Verbindung mit dem ungenauen Ergebnis, was zu erwarten ist, wird diese in der vorliegenden Überprüfung nicht angewendet.

Eine Alternative hierzu besteht in der Nutzung der gering abgedeckten Bereiche des bereits erhobenen Datensatzes als Testdatensatz für diese Anforderung (A1_M2). Besitzt der Datensatz im Fall von Supervised-Learning-Label, wird im Rahmen des Entwicklungsprozesses der zur Verfügung stehende Datensatz zwar ohnehin in Trainings-, Validierungs- und Testdatensatz geteilt, jedoch unterscheiden sich die Anforderungen des im Rahmen der Entwicklung erstellten Testdatensatzes von denen des im Rahmen dieser Robustheitsüberprüfung vorliegenden.[314] Der Testdatensatz, der im Rahmen der Entwicklung erstellt wird, hat einen hohe Repräsentativität des Gesamtdatensatzes aufzuweisen, damit die Leistungsfähigkeit des trainierten Modells ganzheitlich, d.h. über alle Wertbereiche hinweg, be-

[314] Zur Begründung der Teilung des Gesamtdatensatzes siehe Abschnitt 2.2.1.

stimmt wird. Der Testdatensatz, der zur Überprüfung der Anforderung A1 extrahiert wird, enthält hingegen genau die Bereiche, die im Gesamtdatensatz gering repräsentiert sind. Hierdurch ist beim Vorliegen von Labeln im Datensatz ein zusätzlicher vierter Teildatensatz gleich zu Beginn der Entwicklung von den Trainingsdaten zu separieren, mit welchen die Anforderung A1 überprüft wird. Durch das Fehlen von Labeln im Fall von Unsupervised-Learning existiert kein Zweck zur Teilung des Datensatzes gleich zu Beginn der Entwicklung, da die Grundlage der Leistungs- bzw. Vorhersageevaluation fehlt. Hierdurch ist für diese Art der Lernverfahren zur Anwendung der Methode A1_M2 zunächst notwendig, eine Referenz für die Vorhersage der Datenpunkte der niedrig abgedeckten Bereiche zu erzeugen. Eine Möglichkeit besteht in der Vorhersage der Ausgangsgröße durch funktionale Anforderungen. Allerdings lassen diese häufig einen großen Interpretationsspielraum der Ergebnisse zu, wodurch sie keine sichere Referenz zur Bewertung des Modellverhaltens bieten.[315] Daneben ist es möglich, die Vorhersage durch ein Modell zu nutzen, welches zusätzlich auf diesen gering abgedeckten Bereichen, d.h. auf dem gesamten Datensatz, trainiert wurde. Dieses Vorhersage-Modell hat zuvor die funktionalen Anforderungen zu erfüllen, damit es eine ausreichende Vorhersagesicherheit besitzt. Dieses auf allen Daten trainierte Modell wird jedoch nicht als finales Modell genutzt, sondern lediglich das um die gering abgedeckten Bereiche reduzierte. Da jedoch im vorliegenden Anwendungsfall die gering abgedeckten Bereiche der Gesamtdaten nicht vor dem Training des finalen Modells entfernt wurden, wurde das beschriebene, systematisch korrekte Vorgehen nicht angewendet.

Einen möglichen Ausweg für Unsupervised-Ansätze hierzu bietet, analog zu A1_M2, die Reduzierung des vorhandenen Datensatzes um die Datenpunkte von gering abgedeckten Bereichen, wobei das mit dem reduzierten Datensatz trainierte Modell anschließend in seiner Vohersageübereinstimmung gegenüber dem originalen Modell überprüft wird (A1_M3). Die Clusterübereinstimmung bezieht sich hierbei lediglich auf die zum Training genutzten Daten. Liegt eine hohe Übereinstimmung der Clusterzuordnung sowie der Erfüllung der funktionalen Anforderungen durch das reduzierte Modell vor, wird die Clusterübereinstimmung der Datenpunkte, die aus dem Training entfernt wurden, zwischen dem originalen und dem reduzierten Modell überprüft. Zwar wird durch A1_M3 die Erfüllung der Anforderung A1 nicht direkt für das finale Modell überprüft, jedoch wird durch die Sicherstellung des funktional ähnlichen Verhaltens des reduzierten Modells im Vergleich zum originalen Modell eine vergleichbare Basis zur Anforderungsbewertung geschaffen. Liegt keine ausreichende Übereinstimmung zwischen dem reduzierten und dem originalen Modell vor, ist zwar die Anforderung A1 damit noch nicht verletzt, dennoch leitet sich hieraus eine hohe Sensitivität des originalen Modells auf Datenpunkte aus gering abgedeckten Bereichen ab. Zur Anwendung dieser Methode werden verschiedene Schritte definiert, die es zur Anforderungsüberprüfung zu absolvieren gilt.

[315] Siehe Abschnitt 6.2.2.

1. Identifikation der gering abgedeckten Bereiche der Trainingsdaten.
2. Evaluation, ob das reduzierten Modells weiterhin die funktionalen Anforderungen erfüllt und ob die Leistungsfähigkeit des reduzierten Modells hinsichtlich Clustervorhersage auf dem zum Training genutzten Daten (reduzierte Datenmenge) auf gleichem Niveau ist.
3. Evaluation der Leistungsfähigkeit der Vorhersage der gewonnenen Testdaten.

Diese Schritte werden im Folgenden für den Anwendungsfall durchgeführt.

Schritt 1: Evaluation der gering abgedeckten Bereiche

Zur Identifikation gering abgedeckter Bereiche in den Trainingsdaten werden die Häufigkeitsverteilungen der einzelnen Eingangsgrößen analysiert. Die Datenpunkte, deren Werte die geringste Häufigkeit besitzen, werden als potentielle Testdaten ausgewählt. Die finale Festlegung der Anzahl der Testdaten hängt von Schritt 2 ab, weshalb nur von potentiellen Testdaten gesprochen wird.

In Abbildung 6-24 ist die normierte Häufigkeit der Eingangsgröße „maximale Geschwindigkeit“ aufgetragen. Die Bereichsgröße jedes Balkens jeder Eingangsgröße wird so festgelegt, dass Bereiche existieren in denen maximal zwei Datenpunkte vorhanden sind. Das ist notwendig, da aufgrund der unterschiedlichen Wertebereiche der Merkmale keine feste Bereichsgröße festzulegen ist. Durch die Begrenzung der maximal vorkommenden Datenpunkte in mindestens einem Balken bzw. Bereich wird erreicht, dass eine genügend hohe Anzahl an Bereichen zur Identifikation der gering abgedeckten Wertebereiche vorherrscht. Als Ausgangspunkt der Bereichsgröße wird 20 genutzt.

Alle Datenpunkte, die der geringsten Häufigkeit eines Balkens auftreten, werden als potentielle Testdaten festgelegt. Im vorliegenden Fall beträgt die geringste relative Häufigkeit 1,2 % bzw. in absoluten Zahlen ausgedrückt, zwei Datenpunkte. Sie tritt in drei Bereichen auf, wodurch sechs Datenpunkte für die potentielle Reduktion ausgewählt werden.

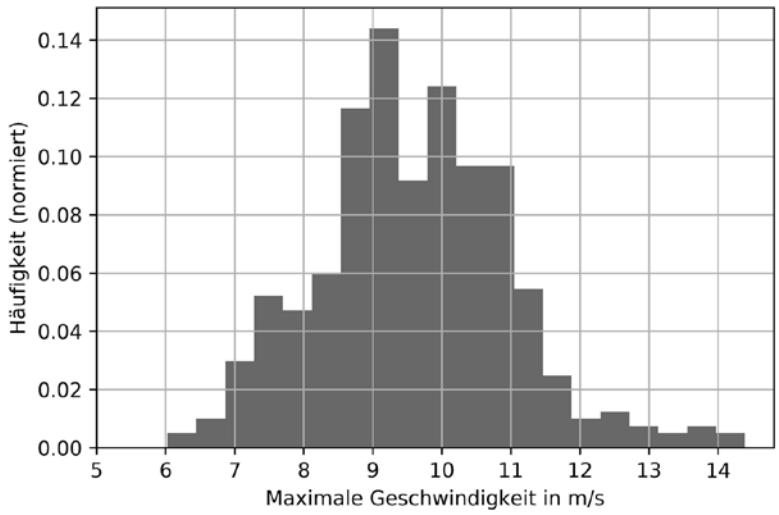


Abbildung 6-24: Histogramm der maximalen Geschwindigkeit

Dieses Vorgehen wird auf alle neun Eingangsmerkmale angewendet.[316] Dabei werden folgende potentielle Datenpunktmengen pro Merkmal ausgewählt:

- Maximale Längsbeschleunigung: 5
- Maximale Querbeschleunigung: 7
- Minimale Querbeschleunigung: 2
- Maximale Lenkradwinkelgeschwindigkeit: 6
- Standardabweichung der Lenkradwinkelgeschwindigkeit: 7
- Minimale Lenkradwinkelgeschwindigkeit: 10
- Maximaler Ruck: 8
- Minimaler Ruck: 9

Die gesamte Menge der potentiellen Testdaten beträgt jedoch anstatt 60 Datenpunkten aus lediglich 51 Datenpunkten, da die gering abgedeckten Bereiche einzelner Merkmale teilweise vom selben Datenpunkt ausgehen.

Schritt 2: Evaluation der funktionalen Anforderungen und der Leistungsfähigkeit

Die Überprüfung der Erfüllung der funktionalen Anforderungen durch ein Modell, welches auf Daten trainiert wurde, deren Menge um die aus Schritt 1 identifizierten 51 Datenpunkte, d.h. auf 88 % der Daten, reduziert wurde, sowie die Evaluation der Leistungsfähigkeit des reduzierten Modells bildet den nächsten Schritt. Im Rahmen der Anforderung DQ1[317] wurden bereits ähnliche Testfälle (Nr. 1 – 3 der Tabelle 6-1) mit negativem Ergebnis hinsichtlich der Anforderungserfüllung, Leistungsfähigkeit und Reproduzierbarkeit durchgeführt. Allerdings reduziert sich die Datenmenge in den DQ1 zugehörigen Testfällen durch zufällige Datenauswahl auf 67 %, woraus sich keine Aussage auf die Erfüllung der funktionalen Anforderungen und die Leistungsfähigkeit bei einer geringeren Reduktion, wie sie im Rahmen der Überprüfung von A1 benötigt wird, treffen lässt. Daher wird zunächst überprüft, ob bereits das Modell, dessen Trainingsdatensatz um die identifizierten 51 Datenpunkte reduziert wurde, die Anforderungen L1 bis L3 erfüllt und eine hohe Übereinstimmung der Clustervorhersage des reduzierten zum originalen Modell erzielt wird. Werden diese Anforderungen und die Leistungsfähigkeit nicht erfüllt, so wird die Menge der reduzierten Datenpunkte verringert und nach den Ursachen dieses Verhaltens gesucht. Die Ergebnisse der resultierenden Testfälle sind in Tabelle 6-6 dargestellt.

---

[316] Die Histogramme der acht verbleibenden Merkmale sind in Anhang D.3 gegeben.

[317] Die gleichmäßige Änderung (bspw. über alle Klassen hinweg) der Datensatzgröße in den unterschiedlichen Entwicklungsphasen bis zu einem gewissen Schwellwert besitzt keine Auswirkungen auf die grundlegende Funktionalität des Modells.

Tabelle 6-6: Testfälle Schritt 2 der Anforderung A1

Nr.	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Reduzierung von**	**Übereinstimmung**
1	nein	nein	ja	51 Datenpunkten (12 %)	42 %
2	-	-	-	35 Datenpunkten (9 %)	42 %
3	ja	ja	ja	34 Datenpunkten (alle außer Ruck)	86 %

Die Reduktion des Trainingsdatensatzes um alle 51 Datenpunkte erzeugt ein Modell, welches die funktionalen Anforderungen L1 und L2 verletzt.[318] Die Übereinstimmung der Clusterzuordnung zwischen reduziertem und originalem Modell des reduzierten Datensatzes liegt bei 42 %. Dies zeigt die hohe Relevanz der entfernten Datenpunkte am funktional korrekten Ergebnis, obwohl sie „nur“ 12 % der Gesamtdatenmenge betragen. Die geringe Leistungsfähigkeit lässt sich ebenfalls an der Darstellung der Akzeptanzwahrscheinlichkeiten der Zeitlücken beim Linksabbiegen zeigen. Wie in Abschnitt 6.1.4 erwähnt, werden die Cluster durch die Auswertung der Akzeptanzwahrscheinlichkeit zu Fahrstilen zugeordnet. Zusätzlich dient die Trennbarkeit der Wahrscheinlichkeitsverläufe als Gütekriterium des Modells. Im Vergleich der Akzeptanzkurven von reduziertem zu originalem Modell in Abbildung 6-25 wird deutlich, dass das reduzierte Modell eine schlechtere Trennbarkeit der Cluster C1 (ausgeglichen) und C2 (sportlich) besitzt.

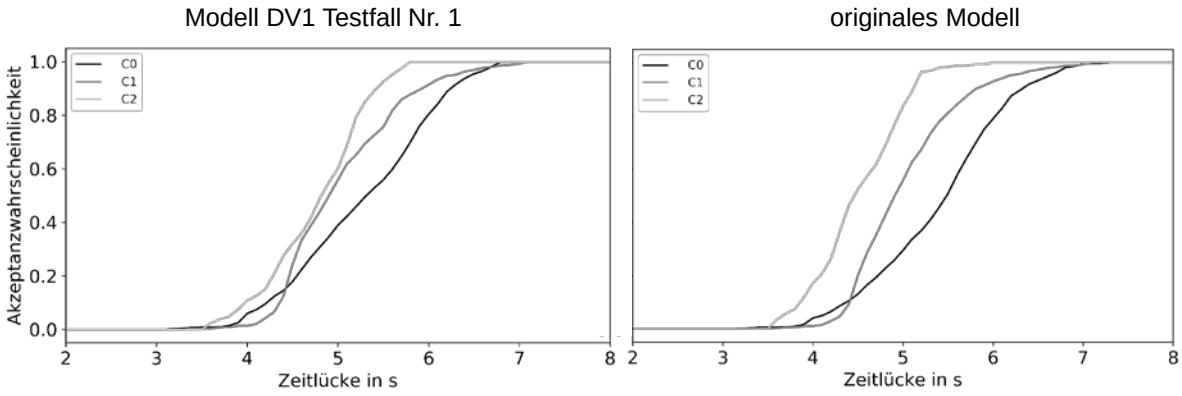


Abbildung 6-25: Vergleich der Akzeptanzkurven

Zur Verbesserung der Leistungsfähigkeit des reduzierten Modells wird die Menge der potentiellen Testdaten von 51 auf 35 verringert. Die Verringerung beruht auf der Entfernung eines Datenpunkts aus den potentiellen Testdaten aus Bereichen jedes Merkmals, die mindestens zwei Datenpunkte enthalten. Hierdurch werden noch immer die Daten aus den Trainingsdaten entfernt, die die geringste Auftretenshäufigkeit besitzen, wodurch es möglich ist, die Anforderung A1 zu überprüfen.

[318] Siehe Anhang 8-30 und Anhang 8-31.

Es ist jedoch nicht möglich, das mit diesen geringer reduzierten Trainingsdaten gelernte Modell (9 % der Daten wurden entfernt) zuverlässig hinsichtlich der Anforderungserfüllung zu bewerten, da die Evaluationsgrundlage der Clusterzuordnung zu Fahrstilen keine Unterscheidung zwischen Cluster C1 und C2 zulässt, wie in Abbildung 6-26 dargestellt. Hierdurch wird die Sensitivität des originalen Modells auf Trainingsdaten von gering abgedeckten Bereichen gezeigt. Allerdings lässt diese keine direkten Rückschlüsse auf die korrekte Funktionalität des originalen Modells in (anderen) gering abgedeckten Bereichen zu, wenn alle zur Verfügung stehenden Daten zum Training genutzt werden. Die Leistungsfähigkeit gemessen mit der Clusterübereinstimmung zwischen reduziertem Modell und originalem Modell verbesserte sich im Vergleich zu Testfall Nr. 1 nicht.

Abbildung 6-26: Akzeptanzkurven des Testfalls Nr. 2 der Anforderung A1

Um ein reduziertes Modell zu erhalten, welches dennoch zur Anforderungsüberprüfung heranzuziehen ist, sind in weiteren Testfällen entweder die potentiellen Testdaten über alle Merkmale, die für die Auswahl in diese Testdatenmenge relevant sind, weiter zu reduzieren oder die potentiellen Testdaten gezielt um einzelne Merkmale zu verringern. Da die erste Möglichkeit zur Folge hat, dass nur noch wenige Testdaten zur tatsächlichen Anforderungsüberprüfung von A1 zur Verfügung stehen und die weitere Reduktion über alle Merkmale, wie in Testfall Nr. 2 gezeigt, bisher keinen Erfolg brachte, wird die zweite Möglichkeit weiterverfolgt. Hierdurch wird ebenfalls implizit die Sensitivität des originalen Modells auf die im Trainingsdatensatz gering abgedeckten Bereiche untersucht. Aufgrund der in Abschnitt 6.2.2.1 festgestellten hohen Relevanz des Rucks als Eingangsgröße werden die potentiellen Testdaten, die aus den Merkmalsverteilungen des maximalen und minimalen Rucks resultieren, wieder als Trainingsdaten genutzt. Hierdurch wird die Menge der potentiellen Testdaten auf 34 verringert. Das resultierende Modell erfüllt die Anforderungen L1 bis L3 und besitzt eine Clusterübereinstimmung von 86 %. Die Sensitivität des originalen Modells ist daher nicht pauschal auf gering abgedeckte Bereiche einzelner Merkmale zurückführbar, sondern auf die der Merkmale minimaler und maximaler Ruck. Mit dem aus diesem Testfall resultierenden Modell ist es möglich, den weiteren Schritt zur Anforderungsüberprüfung A1 durchzuführen.

3. Schritt: Leistungsfähigkeit zur Vorhersage der Testdaten

Der Vergleich der Leistungsfähigkeit zwischen originalem und aus Testfall Nr. 3 resultierenden Modell auf den Testdaten beträgt 91 % bzw. einer Übereinstimmung von 31 der 34 Testdaten in der Clusterzuordnung. Dieses Ergebnis lässt auf eine ausreichende Funktionalität des originalen Modells auch in gering abgedeckten Bereichen der Eingangsmerkmale, außer denen des Rucks, schließen. Für die Merkmale des Rucks wird aufgrund der hohen Sensitivität des Modells auf die Daten aus Bereichen mit einer geringen Abdeckung dieser Schluss nicht gezogen. Ob die bisher im Trainingsdatensatz enthaltenen Daten genügen, um auch in diesen Merkmalen eine ausreichende Funktionalität in gering abgedeckten Bereichen zu garantieren, ist durch die gezielte Erhebung weiterer Testdaten zu beantworten. Die Anforderung A1 wird für die Eingangsmerkmale, außer denen des Rucks, durch das reduzierte Modell erfüllt.

Fazit Anwendbarkeit

Es wurden drei unterschiedliche Möglichkeiten vorgestellt, die Anforderung A1 nach der Funktionalität des gelernten Modells in gering abgedeckten Bereichen der zugehörigen Trainingsdaten zu überprüfen.

Die Möglichkeit der Erhebung zusätzlicher Testdaten (A1_M1) unterliegt hohen Anforderungen hinsichtlich vergleichbarer Bedingungen zur Erhebung des Trainingsdatensatzes. Nur hierdurch ist sichergestellt, dass gezielt die Anforderung A1 überprüft wird und keine Auswirkungen anderer Effekte durch unterschiedliche Bedingungen zwischen Trainings- und Testdaten falsch interpretiert werden. Zusätzlich sind entweder eine hohe Anzahl an Testdaten zu erheben oder gezielt Testfälle zu generieren, um Daten zu erhalten, die in den bisherigen Trainingsdaten gering abgedeckt sind. Beide Verfahren sind mit einem hohen Aufwand verbunden. Des Weiteren besteht die Problematik, dass im Rahmen von Unsupervised-Learning keine Label der Datenpunkte für eine direkte Evaluation der Vorhersageleistung zur Verfügung stehen. Der Abgleich der Vorhersage mit funktionalen Anforderungen oder einem Modell ist hierbei als ungenau einzustufen, da diese meistens ebenfalls nicht auf ihre Validität in Randbereichen überprüft wurden.

Die Anwendung der Methode A1_M2 ist bereits zu Beginn der Entwicklung des gelernten Modells zu berücksichtigen, da diese darauf basiert, dass vor dem Training des finalen Modells die Bereiche, die im Gesamtdatensatz eine geringe Abdeckung besitzen, aus dem Datensatz des Trainings entfernt werden. Es wird erwartet, dass diese Methode eine geringe praktische Anwendung findet, da die Entwicklung eines gelernten Modells zur Erfüllung der ISO 26262[319] personell unabhängig von der Prüfung dessen Sicherheit erfolgt. Allerdings verursacht diese Methode den geringsten Aufwand, wodurch sie zu präferieren ist.

[319] Vgl. ISO: ISO 26262:2018. Road vehicles: Functional safety (2018).

Hierdurch motiviert wird eine alternative Vorgehensweise (A1_M3) vorgeschlagen, die zeigt, dass es möglich ist, auch ohne Erhebung neuer Daten und vorheriger Entfernung gering abgedeckter Bereiche Erkenntnisse bezüglich der Generalisierbarkeit des finalen Modells zu gewinnen. Allerdings gelten diese Erkenntnisse, wie auch in den Methoden A1_M1 und A1_M2 nur für die Bereiche, in denen überhaupt Testdaten vorhanden sind bzw. für die Testdaten erhoben wurden. Eine Begrenzung des Modells im Betrieb auf die überprüften Bereiche ist aus Gründen der funktionalen Sicherheit ratsam.

Die direkte Anwendbarkeit der Anforderungsüberprüfung wird im vorgestellten Anwendungsfall durch das alternative Vorgehen A1_M3 weder festgestellt noch widerlegt. Da allerdings eine Analogie im Vorgehen von A1_M3 zu A1_M2 vorherrscht, wird die Anwendbarkeit der Überprüfung der Anforderung dennoch gezeigt.

Fazit Generalisierbarkeit

Durch die Überprüfung der Anforderung A1[320] wird die Sensitivität des gelernten Modells auf im Trainingsdatensatz selten vorkommende Datenpunkte festgestellt. Wird dabei angenommen, dass der Trainingsdatensatz die späteren Betriebsbedingungen des Modells repräsentiert, bedeutet das eine Überprüfung des funktional korrekten Verhaltens bzw. ausreichende Generalisierbarkeit in selten auftretenden Betriebsbereichen.

Diese selten vorkommenden Datenpunkte stellen im vorliegenden Anwendungsfall häufig die Extremwerte der einzelnen Merkmale dar. Es wird gezeigt, dass bei der Entfernung der gering abgedeckten Bereiche dieser Ruck-Merkmale kein Modell mit korrekter Funktionalität resultiert. Es besteht daher eine hohe Sensitivität auf die gering abgedeckten Bereiche der Merkmale des Rucks des final gelernten Modells. Werden die gering abgedeckten Bereiche der Merkmale des Rucks zum Training genutzt und gleichzeitig die selten auftretenden Werte der anderen Eingangsgrößen aus dem Trainingsdatensatz entfernt, so entsteht ein Modell mit einem funktional korrekten Verhalten, woraus eine vergleichsweise geringe Sensitivität des finalen Modells auf die anderen Merkmale gefolgert wird. Durch die hohe Vorhersageleistung der seltenen Datenpunkte des reduzierten Modells, gemessen an der Vorhersage durch das finale Modell, wird dem reduzierten Modell eine ausreichende Generalisierbarkeit auch in den gering abgedeckten Bereichen der Merkmale außer denen des Rucks zugeschrieben.

6.2.3.4 Trainingsprozess T1

- T1: Der Initialisierungsprozess der Algorithmen führt zu Modellen, die die grundlegend gleiche Funktionalität besitzen.

Zur Überprüfung dieser Anforderung werden die Initialisierungsparameter im Initialisierungsprozess verändert und hiermit ein neues Modell trainiert. Zur Evaluierung stehen ver-

[320] Die Funktion des Modells ist auch in dessen Bereichen mit einer spärlichen Abdeckungsrate durch die Trainingsdaten gewährleistet.

schiedene Methoden zur Verfügung: Der direkte Vergleich der Modellstruktur zwischen den Modellen unterschiedlicher Parametrierung, was im Anwendungsfall der Vergleich der Lage der Clusterschwerpunkte entspricht, stellt eine Möglichkeit dar. Bei Übereinstimmung der Schwerpunkte bzw. der Modellparameter ist die Anforderung T1 komplett erfüllt. Existieren Abweichungen der Parameter, ist es möglich die grundlegend gleiche Funktionalität der Modelle zu überprüfen, indem die Erfüllung der funktionalen Anforderungen überprüft und die resultierende Vorhersagen einzelner Datenpunkte verglichen werden. Der Vergleich der Vorhersage findet dabei entweder relativ zwischen dem originalen und dem neu initialisierten Modell statt (Unsupervised-Learning) oder zwischen dem neu initialisierten Modell und den Labeln des Datensatzes (Supervised-Learning).

In der für den Anwendungsfall genutzten Implementierung des K-Means-Algorithmus ist bereits ein Initialisierungsprozess implementiert, welcher die bereits in Unterkapitel 5.5 erwähnte *n*-fache Initialisierung durchführt und aus der resultierenden Modellschar das Optimum auswählt.[321] Die Initialisierung wird dabei entsprechend des K-Means++ Algorithmus[322] durchgeführt, da dieser eine effizientere Berechnung des K-Means-Modells bei gleichzeitigem Erreichen eines geringeren verbleibenden Fehlers ermöglicht. Dieser wählt die Clusterschwerpunkte wie folgt aus:

1. Zufällige Auswahl eines ersten Clusterschwerpunkts
2. Berechnung der Abstände zwischen allen Datenpunkten und den am nächsten gelegenen Clusterschwerpunkten (soweit bereits vorhanden)
3. Zufällige Auswahl eines neuen Datenpunkt als neues Clusterschwerpunkts unter der Bedingung, dass dieser aus der Datenmenge stammt, die eine hohe Distanz zu den bestehenden Datenpunkten besitzt.
4. Wiederholung von Schritt 2. und 3. bis alle benötigten Clusterschwerpunkte initial festgelegt wurden.

Hieran schließt sich das Vorgehen des „normalen“ K-Means an. Aufgrund der Nutzung dieses Initialisierungsprozesses ist zur Überprüfung der Anforderung lediglich der Lernprozess inkl. der Initialisierung mehrmals durchzuführen, da durch die Zufallsauswahl des ersten Clusterschwerpunkts bereits die Variation durchgeführt wird, die bei einer zu geringen Anzahl n_{init} an zufälligen Initialisierungen zu einem unterschiedlichen funktionalen Modellergebnis führt. Die *n*-fache Initialisierung wird im Rahmen eines K-Means-Algorithmus genutzt, da dieser Algorithmus abhängig von der Wahl der Initialisierungsparameter in lokalen Optima resultiert.[323] Ohne diesen Initialisierungsprozess wird daher die Anforderung T1 verletzt.

---

[321] Vgl. scikit-learn: sklearn.cluster.KMeans (2019).

[322] Arthur, D.; Vassilvitskii, S.: k-means++: the advantages of careful seeding (2007).

[323] Vgl. Huang, Z.: Clustering Large Data Sets with Mixed Numeric and Categorical Values (1997), S. 27.

Die für das Training des K-Means gewählte Anzahl an Initialisierungen beträgt 70. In jedem Testfall wurde der Initialisierungsprozess mit $n_{init} = 70$ neu ausgeführt. Die Ergebnisse der Testfälle sind in Tabelle 6-7 und Tabelle 6-8 dargestellt.

Tabelle 6-7: Koordinaten der Clusterschwerpunkte der Testfälle der Anforderung T1

Testfall Nr./Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1/0	-0,314	-0,154	-0,603	0,509	-0,643	-0,882	0,790	0,012	0,037
2/0	-0,295	-0,138	-0,556	0,511	-0,640	-0,858	0,763	0,048	0,028
3/0	-0,318	-0,158	-0,596	0,509	-0,641	-0,881	0,786	0,006	0,048
4/0	-0,280	-0,123	-0,567	0,520	-0,654	-0,878	0,782	0,054	0,055
5/0	-0,294	-0,143	-0,565	0,514	-0,644	-0,866	0,774	0,037	0,038
1/1	-0,222	-0,476	0,200	-0,474	0,304	0,519	-0,540	-0,431	0,381
2/1	-0,215	-0,467	0,219	-0,474	0,307	0,518	-0,554	-0,424	0,380
3/1	-0,217	-0,474	0,199	-0,479	0,309	0,527	-0,545	-0,428	0,381
4/1	-0,224	-0,496	0,176	-0,457	0,272	0,495	-0,511	-0,426	0,382
5/1	-0,211	-0,472	0,211	-0,474	0,305	0,522	-0,544	-0,423	0,388
1/2	0,867	1,000	0,686	-0,104	0,589	0,651	-0,465	0,654	-0,668
2/2	0,913	1,044	0,668	-0,155	0,676	0,726	-0,489	0,625	-0,705
3/2	0,867	1,000	0,686	-0,104	0,589	0,651	-0,465	0,654	-0,668
4/2	0,848	1,018	0,696	-0,159	0,689	0,715	-0,524	0,598	-0,700
5/2	0,878	1,034	0,670	-0,146	0,657	0,701	-0,498	0,630	-0,707

Tabelle 6-8: Funktionale Überprüfung der Testfälle der Anforderung T1

Nr.	**L1 erfüllt**	**L2 erfüllt**	**L3 erfüllt**	**Relative Übereinstimmung**	**Datenpunkte mit differierender Vorhersage zu Modell Nr. 1**
1	ja	ja	ja	-	-
2	ja	ja	ja	97 %	63, 182, 246, 261, 281, 291, 324, 329, 362
3	ja	ja	ja	100 %	329
4	ja	ja	ja	97 %	16, 20, 63, 87, 108, 182, 261, 281, 362
5	ja	ja	ja	99 %	63, 182, 261, 281, 329, 362

Für Testfall Nr. 1 stehen noch keine Vergleichswerte zur Verfügung, weshalb aus diesem keine Aussage zur Erfüllung der Anforderung T1 getroffen wird. Der Testfall Nr.1 bzw. das hieraus resultierende Modell wird im Folgenden als Vergleichsmodell herangezogen und stellt das Modell dar, welches in den bisherigen Überprüfungen der Robustheitsanforderungen als das „originale“ bzw. „finale“ Model bezeichnet wurde. Die Clusterschwerpunkte aller Cluster differieren zwischen Testfall Nr. 1 und 2 (siehe Tabelle 6-7), wodurch gezeigt wird, dass die Anzahl an zufälligen Initialisierungen von $n_{\text{init}} = 70$ nicht ausreicht, um reproduzierbare Clusterschwerpunkte hinsichtlich ihrer exakten Lage zu erhalten. Daher sind einerseits weitere Testfälle zur Analyse notwendig und andererseits gilt es ebenfalls die Funktionalität des Modells zu untersuchen, um die Auswirkungen der Veränderung der Clusterschwerpunkte zu bewerten. Bei einer Übereinstimmung der Clusterschwerpunkte wären lediglich weitere Testfälle zur Überprüfung der Reproduzierbarkeit notwendig sowie eine Auswertung der Lage dieser Punkte ausreichend. Die Auswertung des Testfalls Nr. 2 hinsichtlich der funktionalen Anforderungen und der Übereinstimmung der Vorhersage zwischen Testfall Nr. 1 und 2 zeigt, dass L1 bis L3 erfüllt werden und eine Übereinstimmung von 97 % erreicht wird. Es wird neben der quantitativen Erfassung der Übereinstimmung auch analysiert, in welchen Datenpunkten sich die Vorhersagen unterscheiden, um zu untersuchen, ob die Unterschiede immer in den gleichen Datenpunkten auftreten. Dies könnte einerseits darauf hinweisen, dass das Vergleichsmodell aus Testfall Nr. 1 ungünstig gewählt ist und die anderen Modelle der übrigen Testfälle zueinander ggf. eine höhere Ähnlichkeit aufweisen als zum Vergleichsmaßstab. Andererseits ist es möglich, dass die Datenpunkte, die sich häufig unterscheiden, an direkten Clustergrenzen liegen und daher sensitiv auf kleine Änderungen der Lage der Clusterschwerpunkte reagieren. In Testfall Nr. 3 ist der Schwerpunkt des Cluster C2 identisch mit dem des Falls Nr. 1. Die Schwerpunkte C1 unterscheiden sich leicht. Bis auf einen Datenpunkt sind die Vorhersagen des resultierenden Modells identisch, wodurch eine Erfüllung der Anforderungen L1 bis L3 erreicht wird. Die Clusterschwerpunkte des Testfalls Nr. 4 differieren alle vom Vergleichsmodell. Es wird die gleiche Übereinstimmungsrate wie in Testfall Nr. 2 erreicht. Die aus Testfall Nr. 5 resultierende Vorhersage unterscheidet sich in sechs Datenpunkten von Nr. 1. Auch hier sind alle Clusterschwerpunkte im Vergleich leicht verändert.

Kein Datenpunkt besitzt in allen vier Testfällen eine veränderte Vorhersage im Vergleich zum Modell des Testfall Nr. 1. Der Datenpunkte 63, 182, 261, 281, 329 und 362 rufen in drei der vier Testfällen eine veränderte Vorhersage vor, weshalb sie näher analysiert werden. Wie in Tabelle 6-8 dargestellt, treten fünf der sechs Datenpunkte gemeinsam in drei Testfällen (Nr. 2, 4 und 5) auf. Diese Modelle sind sich hinsichtlich ihrer Vorhersage zueinander ähnlicher als zum Vergleichsmaßstab. Dennoch ist nicht von einer ungünstigen Wahl des Vergleichsmaßstabs auszugehen, da einer der vier Testfälle in einem Modell mit sehr hoher Ähnlichkeit resultiert. Alle betreffenden Datenpunkte befinden sich in der Darstellung der funktionalen Anforderungen L1 bis L3 in Wertebereichen, die zu unterschied-

lichen Fahrstilen zuordenbar sind.[324] Auch die Darstellung der betreffenden Datenpunkte in einer PCA[325] zeigt, dass sich direkt im Grenzbereich zweier Cluster befinden.[326] Daher wird durch die unterschiedliche Zuordnung je nach Auswahl der Initialisierung kein funktional falsches Verhalten hervorgerufen.

Aufgrund der hohen Clusterübereinstimmung, des funktional korrekten Verhaltens aller aus den Testfällen resultierenden Modelle und der Feststellung, dass die sich ergebenden Änderungen auf die minimale Verschiebung von Clusterschwerpunkten zurückzuführen sind, wird die Anforderung T1 als erfüllt angesehen. Die Generalisierbarkeit des originalen Modells ist daher in dieser Hinsicht als ausreichend zu bewerten.

Eine weitere Erhöhung der Anzahl an zufälligen Initialisierungen ist im Fall des vorliegenden Algorithmus durch den geringen Rechenaufwand möglich und verursacht eine noch geringere Streuung der Clusterschwerpunkte. Ebenfalls ist es möglich, den Schwellwert der Differenzen zwischen den gleichen Clusterpunkten zwei aufeinanderfolgender Iterationsschritte, der zum Stopp des Algorithmus führt, zu verkleinern, um eine geringe Streuung der Clusterschwerpunkte zu erhalten. Jedoch wird darauf verzichtet, da sich hierdurch keine Leistungssteigerung des Modells gewährleistet ist, da die Bewertung der Leistung des Modells ohne die Kenntnis der Ground-Truth erfolgt. Das Weglassen einzelner Datenpunkte besitzt z.B. einen höheren Einfluss auf das relative Modellergebnis als die vorliegende Streuung der Clusterschwerpunkte.[327] Mit einer „Stabilisierung“ der Clusterschwerpunkte wird daher keine Verbesserung der Generalisierbarkeit bzw. des Modellergebnisses erreicht, sondern nur der Aufwand zur Berechnung im Training erhöht.

Fazit Anwendbarkeit

Die Anwendbarkeit der Anforderung T1[328] wurde erfolgreich gezeigt. Es handelt sich bei der vorgestellten Methode zur Evaluierung der Anforderungserfüllung lediglich um einen Ansatz, der auf zufälliger Auswahl der Überprüfungspunkte basiert. Hierdurch wird zwar nicht der ungünstigste Fall der Wahl der Initialisierungen überprüft, was jedoch durch die *n*-fache Initialisierung des Algorithmus in der K-Means++-Implementierung nicht notwendig ist. Selbst wenn die *n*-fache Initialisierung nur aus „schlechten“ Initialwerten des ersten Clusters bestehen, werden durch die K-Means++-Implementierung die weiteren Cluster „intelligent“ gewählt, wodurch der am schlimmsten mögliche Fall nicht eintritt.

Die Überprüfung der Anforderung auf rechenintensive Modelle bleibt als offene Fragestellung bestehen. Für den vorliegenden Anwendungsfall wurden insgesamt 350 unterschiedl-

[324] Siehe Anhang 8-32 bis Anhang 8-34.

[325] Siehe Abschnitt 6.2.3.2, Methode DV1_M1.

[326] Siehe Anhang 8-35.

[327] Siehe Abschnitt 6.2.3.1, Anforderung DQ1 und DQ2.

[328] Der Initialisierungsprozess der Algorithmen führt zu Modellen, die die grundlegend gleiche Funktionalität besitzen.

iche Initialisierungen durchgeführt, d.h. 350 Modelle wurden trainiert, um festzustellen, dass die Anzahl an zufälligen Initialisierungen für ein funktional robustes Modellergebnis genügt. Allerdings ist der Initialisierungsprozess für eine statistisch signifikante Aussage analog zu den Berechnungen in DQ1[329] 59 Mal durchzuführen, wodurch 3990 Modelle zu trainieren sind. Selbst wenn innerhalb des Initialisierungsprozesses lediglich ein Modell ausgewählt wird, ist beispielsweise im Fall von Neuronalen Netzen eine Durchführung von 59 Initialisierungen abhängig von den zur Verfügung stehenden Ressourcen ggf. nicht praktikabel, da ein Zeitaufwand von mehreren Wochen benötigt würde. Zusätzlich ist die Anzahl an notwendigen Initialisierungen nach oben nicht begrenzt, sondern hängt von den Resultaten ab, die durch diese Testfälle entstehen. In der Anwendbarkeit der Überprüfung dieser Anforderung auf rechenintensive Modelle besteht noch weiterer Forschungsbedarf.

Fazit Generalisierbarkeit

Durch die Überprüfung der Anforderung wird festgestellt, wie sensitiv die erlernten Zusammenhänge auf den gewählten Initialisierungsprozess sind bzw., ob lokale statt dem globalen Optimum erreicht werden. Im vorliegenden Anwendungsfall wird festgestellt, dass sich die durch den Initialisierungsprozess resultierenden Modelle zwar in ihrer Struktur voneinander unterscheiden, d.h. unterschiedliche lokale Optima erreicht werden, jedoch das funktionale Verhalten hiervon nicht beeinflusst wird. Hierdurch wird eine durch den Initialisierungsprozess sichergestellte ausreichende Generalisierbarkeit des Modells bestätigt, obwohl der Algorithmus eine hohe Sensitivität auf die Wahl der Initialisierungsparameter besitzt.

Die Analyse der Anforderung T1 ist geeignet, um die erreichte Generalisierbarkeit hinsichtlich ihrer Stabilität gegenüber Änderungen im Initialisierungsprozess zu untersuchen. Eine geringe Stabilität des Prozesses und vor allem hieraus resultierendes funktional falsches Modellverhalten weist auf das Erreichen von lokalen Optima hin, in denen eine geringere Generalisierbarkeit erwartet wird. Im Rahmen der Überprüfung der anderen Robustheitsanforderungen, die ein erneutes Training des Modells mit einem veränderten Datensatz erfordern, ist daher sicherzustellen, dass im Vergleich zum originalen Modell ein ähnliches Optimum erreicht wird. Ist dies nicht der Fall, sind die Erkenntnisse die hinsichtlich der Generalisierbarkeit aus diesem erneuten Training gewonnen werden, nicht oder nur teilweise auf die eingebrachte Veränderung der Trainingsdaten zurückzuführen, da das veränderte Optimum ebenfalls die Generalisierbarkeit ändert.

6.3 Fazit

Der Ansatz zur Identifikation fehlender Generalisierbarkeit wurde in den Schritten drei (Überprüfung der funktionalen Anforderungen) und vier (Überprüfung der Robustheitsan-

[329] Siehe Anhang D.6.

forderungen) auf den Anwendungsfall der Fahrstildetektion als Teil eines Fahrerassistenzsystems angewendet. Die Anwendung der ersten beiden Schritte (Handlungsempfehlungen zur Sicherstellung der Qualität von Daten, Methoden und Prozesse sowie direkte Überprüfung von Ursachen fehlender Generalisierbarkeit) wurde hierbei nicht prototypisch überprüft, da diese lediglich optional sind.

In den folgenden Abschnitten wird jeweils auf das Fazit der Anwendung und der gewonnenen Erkenntnisse hinsichtlich der Generalisierbarkeit für den Anwendungsfall für jeden Schritt einzeln eingegangen. In Abschnitt 6.3.3 wird anschließend zusammenfassend die Forschungsfrage nach der Anwendbarkeit des Ansatzes beantwortet.

6.3.1 Fazit der funktionalen Anforderungen

Die Anforderungen L1 – L4 wurden bei zwei unterschiedlichen Konfigurationen der Eingangsgrößen des gelernten Modells der Fahrstildetektion überprüft. L1 fordert eine Ordinalität der identifizierten Fahrstile bezüglich der Geschwindigkeit bzw. Längsbeschleunigung, L2 bezüglich der Längs- bzw. Querbeschleunigung, L3 bezüglich der Lenkradwinkelgeschwindigkeit und L4 bezüglich des Rucks. Im Rahmen der Überprüfung wird gezeigt, welche hohe Sicherheitsrelevanz diese Überprüfung besitzt, vor allem bei Unsupervised-Ansätzen. Durch diese Überprüfung ist es möglich, funktional falsches Verhalten eines gelernten Modells zu identifizieren, auch wenn bisherige Evaluationsmethoden des Modells dieses Fehlverhalten nicht aufdecken. Dies liegt im vorliegenden Anwendungsfall darin begründet, dass die Evaluation nur eine nicht exakt spezifizierte Richtlinie eines plausiblen Modellverhaltens anhand von Funktionsverläufen darstellt und lediglich einen relativen Vergleich zweier Modelle hinsichtlich ihrer Funktionalität bzw. Leistungsfähigkeit erlaubt. Entspricht eine funktionale Anforderung nicht den im Modell verorteten Zusammenhängen, ist jedoch stets nachzuvollziehen, ob den gelernten Zusammenhängen oder der funktionalen Anforderung zu misstrauen ist. Bei Supervised-Ansätzen liegt eine Ground-Truth vor, wodurch die Funktionalität zwischen allen Eingangsgrößen und der/den Ausgangsgröße(n) bereits überprüft wird. Dennoch werden durch die Überprüfung der funktionalen Anforderungen auch in diesem Fall relevante Erkenntnisse gewonnen, da beispielsweise die durch das Modell erlernten Zusammenhänge zwischen den einzelnen Eingangsgrößen überprüft werden. Da das Modell die Vorhersage der Ausgangsgröße durch die Kombination der zur Verfügung stehenden Eingangsgrößen vornimmt, ist es möglich, dass einzelne Eingangsgrößen funktional inkorrekte Zusammenhänge zur Ausgangsgröße enthalten, obwohl durch die Kombination der Eingangsgrößen insgesamt das korrekte Ergebnis resultiert.

Im vorliegenden Fall erfüllen die Zusammenhänge des finalen Modells drei der vier Anforderungen, was darauf hinweist, dass die verletzte Anforderung hinsichtlich ihrer Anwendbarkeit bzw. Korrektheit im vorliegenden Fall zu untersuchen ist. Eine Analyse der Trainingsdaten zeigt, dass diese Anforderung aufgrund von äußeren Gegebenheiten tatsächlich anzuzweifeln ist. Hierdurch wird gezeigt, welchen Vorteil gelernte Modelle ge-

genüber konventionell programmierten Modellen besitzen. Konventionelle Modelle werden häufig durch die Implementierung von, aus der Literatur bekannten, Zusammenhängen, die durch Realdaten parametriert werden, umgesetzt, was im vorliegenden Fall zu einem funktional falschen Verhalten des konventionellen Modells führen würde.

Verletzt das gelernte Modell mehrere oder alle Anforderungen, wie durch die Konfiguration des gelernten Fahrstilmodells ohne Nutzung der Eingangsgröße Ruck, ist zu überprüfen, ob das gelernte Modell funktional korrekte Zusammenhänge abbildet oder wie im vorliegenden Fall, fehlerhafte Beziehungen aufgrund von fehlenden Eingangsinformationen enthält.

Die Feststellung des funktional korrekten Verhaltens des Modells bezieht sich lediglich auf die im Datensatz abgedeckten Bereiche des Modells. Die Feststellung der darüberhinausgehenden Funktionalität wird durch Überprüfung der Generalisierbarkeit durch die Robustheitsanforderungen adressiert. Beispielsweise wird die Generalisierbarkeit in gering abgedeckten Bereichen des zur Verfügung stehenden Datensatzes in der Anforderung A1 untersucht.

Die Hypothese der Anwendbarkeit des dritten Schrittes des in Unterkapitel 5.1 abgeleiteten Ansatzes zur fehlenden Generalisierbarkeit wurde anhand obiger Anwendung nicht falsifiziert. Zusätzlich zum finalen Stand der Auslegung des Anwendungsfalls, welche lediglich die Zuordnung eines Eingangsdatenpunkts zum stärksten Cluster berücksichtigt und hieraus den Gesamtfahrstil bildet,[330] wird in Anhang C die Anwendbarkeit des dritten Schrittes bei Vorliegen mehrerer möglicher Ausgangsgrößen (Auslegungsvariante B des Anwendungsfalls), gezeigt.

Als eine der Herausforderungen in der Anwendung wird die Definition von geeigneten funktionalen Anforderungen identifiziert, da ML vor allem in den Problemstellungen angewendet wird, die nicht vollständig spezifizierbar sind. Besonders im Rahmen von end-to-end gelernten Modellen, d.h. Modelle, die einen breiten Funktionsumfang, wie bspw. von der Sensorverarbeitung bis zur Ansteuerung von Fahrzeugaktorik, besitzen, ist die Definition von funktionalen Anforderungen schwierig. Durch diesen breiten Funktionsumfang ist einerseits die Verkettung einzelner Anforderungen notwendig, andererseits sind für jede Situation spezifische Anforderungen zu finden, wie genau funktional korrektes Verhalten definiert ist. Durch die Auslegung von Salay et al.[331], dass ML nur auf Komponentenebene einzusetzen ist, um mit der ISO 26262 in Einklang zu stehen, wird diesem Problem der Anwendbarkeit jedoch teilweise Einhalt geboten.

---

[330] Auslegung A, siehe Abschnitt 6.1.4.

[331] Salay, R. et al.: An Analysis of ISO 26262 (2017).

6.3.2 Fazit der Robustheitsanforderungen

Die Erfüllung der in Unterkapitel 5.5 definierten Robustheitsanforderungen wurde bis auf T2, welche fordert, dass Veränderungen der Datensequenzen während des Trainingsprozesses keine Auswirkungen auf die grundlegende Funktionalität des Modells besitzen, durch den vorliegenden Anwendungsfall überprüft. Die Anforderung T2 wird durch den Anwendungsfall prinzipbedingt immer erfüllt, weshalb keine explizite Überprüfung stattfand. Durch die Überprüfung der Robustheitsanforderungen wird gezeigt, dass hierdurch Erkenntnisse über die vorliegende Generalisierbarkeit des finalen Modells gewonnen werden. Diese Erkenntnisse dienen entweder der Verbesserung des Modells, der Bestätigung des Modells oder der Begrenzung des Betriebsbereichs des Modells, damit dieses das übergeordnete Sicherheitsziel nicht verletzt. Eine Übersicht über die Robustheitsanforderungen mit zugehörigen Methoden zur Überprüfung der Anforderungen ist durch Tabelle 6-9 gegeben.

Die Anforderungen bezüglich der Datenquantität DQ1 und DQ2 beruhen auf einer Datensatzreduktion und dem erneuten Training eines Modells auf diesem reduzierten Datensatz. Aus DQ1 wird die Erkenntnis gewonnen, dass im Anwendungsfall die interindividuellen Unterschiede zur Identifikation der relevanten Zusammenhänge zur Fahrstildetektion eine höhere Relevanz besitzen als die intraindividuellen Unterschiede. Hierdurch ist es möglich, im Rahmen von weiteren Datenerhebungen gezielt eine höhere Interindividualität zu erfassen, um eine höhere Generalisierbarkeit des Modells zu erhalten. DQ2 zeigt, dass das Vorliegen von Datenpunkten in Cluster C0 und C2 relevant ist, um die jeweils anderen Cluster voneinander zu trennen. Dies weist auf fehlende Generalisierbarkeit hin. Vor allem die Eingangsgröße der Lenkradwinkelgeschwindigkeit gilt es detaillierter hinsichtlich der Relevanz bzw. potentieller Verbesserung ihrer derzeit genutzten Eingangsmerkmale zu untersuchen, da festgestellt wird, dass das Merkmal der maximalen Lenkradwinkelgeschwindigkeit keine Trennbarkeit zwischen den nicht-reduzierten Clustern besitzt. Beide Anforderungen sind anwendbar, erfordern jedoch das mehrmalige erneute Training des Modells. Im Anwendungsfall wurden 37 neue Modelle zur Anforderungsüberprüfung trainiert, wobei diese Zahl stark von der Anzahl der Klassen/ Cluster des Modells abhängt. Bei rechenintensiven Modellen wird hierdurch ein hoher zeitlicher Aufwand verursacht. Andere Alternativen zur Anforderungsüberprüfung, die kein erneutes Training des Modells erfordern, existieren nicht.

Tabelle 6-9: Übersicht der Robustheitsanforderungen und zugehörige Methoden

Nr.	Anforderung	Methoden-bez.	Kurzbeschreibung	Anwen-dung
DQ1	Die gleichmäßige Änderung (bspw. über alle Klassen hinweg) der Datensatzgröße in den unterschiedlichen Entwicklungsphasen bis zu einem gewissen Schwellwert besitzt keine Auswirkungen auf die grundlegende Funktionalität des Modells.	-	Zufällige Reduktion der Datenpunkte aller Cluster	ja
		-	Gezielte Reduktion von einzelnen Fahrern	ja
		-	Gezielte Reduktion von einzelnen Fahrten pro Fahrer	ja
DQ2	Alle beabsichtigten Klassen sind innerhalb des Trainingsdatensatzes für die grundlegende Funktionalität des Modells hinreichend vertreten. Die Veränderung der Klassenrepräsentanz einzelner Klassen verändert die Leistungsfähigkeit für jede andere beabsichtigte Klasse nicht.	-	Zufällige Reduktion der Datenpunkte eines Clusters	ja
DV1	Mikroskopische Veränderungen der Vorverarbeitung der Eingangsdaten des Modells besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells.	DV1_M1	Änderung der Vorverarbeitung des bestehenden Trainingsdatensatzes	ja
		DV1_M2	Erhebung neuer Testdaten mit veränderter Vorverarbeitung	nein
		DV1_M3	Entfernung der benötigten Testdaten vor eigentlichem Entwicklungsprozess und Veränderung der Vorverarbeitung	nein
		DV1_M4	Nutzung eines bereits auf einem reduzierten Datensatz trainierten Modell mit ähnlicher Leistungsfähigkeit und Veränderung der Vorverarbeitung der übrigen ungesehenen Daten	ja
		DV1_M5	Training eines Modells mit veränderter Vorverarbeitung des originalen Datensatzes	ja
A1	Die Funktion des Modells ist auch in dessen Bereichen mit einer spärlichen Abdeckungsrate durch die Trainingsdaten gewährleistet.	A1_M1	Erhebung neuer Testdaten aus gering abgedeckten Bereichen	nein
		A1_M2	Entfernung der benötigten Testdaten vor eigentlichem Entwicklungsprozess	nein
		A1_M3	Nutzung eines bereits auf einem um gering abgedeckte Bereiche reduzierten Datensatz trainierten Modell mit ähnlicher Leistungsfähigkeit und den ungesehenen Daten als Testdaten	ja
T1	Der Initialisierungsprozess der Algorithmen führt zu Modellen, die die grundlegend gleiche Funktionalität besitzen.	-	Mehrfache Durchführung des Initialisierungsprozesses	ja
T2	Veränderungen der Datensequenzen während des Trainingsprozesses besitzen keine Auswirkungen auf die grundlegende Funktionalität des Modells.	-	Zufällige Permutation der Datensequenzen	nein

Zur Überprüfung von DV1, in welcher die Datenvorverarbeitung mikroskopisch geändert wird, werden fünf verschiedene Methoden identifiziert, wobei drei Methoden (DV1_M1, DV1_M2 und DV1_M3) auf der reinen Generierung von Testdaten basieren. Hierdurch ist die Anwendbarkeit auch bei rechenintensiven Modellen gegeben. Jedoch wird durch eine der zwei Methoden zur Anforderungsüberprüfung, die auf dem Training eines neuen Modells beruht, eine andere Art der Erkenntnis der Generalisierbarkeit gewonnen als in den übrigen Methoden. Innerhalb dieser Methode (DV1_M5) werden Erkenntnisse über die Übertragbarkeit der Modellstruktur auf ähnliche Problemstellungen gewonnen, was einen anderen Aspekt der Generalisierbarkeit des Modells beleuchtet als die Erkenntnisse über die Stabilität der Modellvorhersagen des Modells bei mikroskopisch veränderten Daten, wie sie in den übrigen Methoden erreicht wird. Jedoch genügt für DV1_M5 eine geringere Anzahl an neu trainierten Modellen, als beispielsweise DQ2. Die Anzahl an neu trainierten Modellen hängt lediglich von der zu erreichenden Erkenntnistiefe ab und nicht von Modellstruktur, Anzahl an Eingangsgrößen o.ä.. Die aus DV1_M1 und DV1_M2 gewonnene Erkenntnis hinsichtlich der Generalisierbarkeit ist, dass das originale Modell robust auf eine mikroskopische Änderung der Datenpunkte reagiert. Auch eine teilweise makroskopische Änderung einzelner Merkmale (im vorliegenden Fall zwei von neun Merkmalen) resultiert in einer robusten Vorhersage, was allerdings erst in Zusammenhang mit der Überprüfung DV1_M5 bemerkt wurde. Die weitere Erkenntnis aus DV1_M5 hinsichtlich der Übertragbarkeit der Modellstruktur lautet, dass die Auswahl der Eingangsmerkmale sowie die Wahl der Vorverarbeitung nicht pauschal auf ähnliche Problemstellungen übertragbar sind.

Zur Überprüfung der Anforderungserfüllung A1 (funktional korrektes Verhalten in Bereichen, die durch Daten gering abgedeckt sind) werden drei Methoden vorgestellt, wobei zwei (A1_M1 und A1_M2) auf der reinen Erzeugung von Testdaten, die im Betrieb selten vorkommen, basieren. A1_M1 sieht die Erhebung der Testdaten durch eine zusätzliche Datenerhebung vor, A2_M2 besteht darin, die benötigten Testdaten bereits zu Beginn der Entwicklung des finalen Modells aus dem Gesamtdatensatz zu entfernen. Dadurch, dass im vorliegenden Anwendungsfall diese beiden Methoden nicht angewendet werden konnten, findet eine alternative Methode A1_M3 Anwendung, die zwar nicht direkt die Anforderung A1 überprüft, jedoch eine hohe Übertragbarkeit der Erkenntnisse hinsichtlich der Generalisierung besitzt Dies liegt darin begründet, dass die Überprüfung auf Sensitivität der gering abgedeckten Bereiche auf ein reduziertes, dem originalen ähnlichen, Modell angewendet wird und nicht auf das originale Modell selbst. Hierdurch wird die direkte Anwendbarkeit der Durchführung durch A1_M3 weder bestätigt noch widerlegt. Da allerdings das Vorgehen bis auf die Differenz der Modelle analog zu A1_M2 ist, wird hieraus eine Anwendbarkeit der Anforderung abgeleitet. Durch A1_M3 wird identifiziert, dass eine hohe Sensitivität auf die gering abgedeckten Bereiche der Merkmale des Rucks des finalen Modells vorliegt. Alle anderen Merkmale werden trotz ihrer geringen Abdeckung in den Trainingsdaten robust vorhergesagt. Die Aussagekraft dieser Anforderung ist auf die jeweilig überprüften Datenpunkte begrenzt. Eine Begrenzung des Betriebsbereichs auf die als funktional korrekt deklarierten Bereiche ist als Resultat dieser Anforderungsüberprü-

fung hinsichtlich der Sicherheit des Modells sinnvoll. Hierdurch wird die Verletzung des übergeordneten Sicherheitsziels durch Auftreten von Betriebsbereichen, die nicht im Rahmen von A1 überprüft wurden, verhindert.

Die angewendete Methode zur Überprüfung der Erfüllung der Anforderung T1 beruht auf einer mehrfachen Initialisierung des Modells mit unterschiedlichen Modellparametern. Sie dient der Feststellung, ob ein lokales oder ein globales Optimum durch das Modell erreicht wird. Im vorliegenden Fall wurden aufgrund der hohen Sensitivität des Algorithmus auf die Initialparametrisierung und dem hierdurch ohnehin implementierten Initialisierungsprozess eine Anzahl von 350 Modellen zur Anforderungsüberprüfung trainiert.[332] Durch die kurze Berechnungszeit des Anwendungsfalls ist dieses Vorgehen möglich. Es wurden keine alternativen Vorgehensweisen identifiziert, weshalb die Frage nach Anwendbarkeit dieser Anforderung bei rechenintensiveren Modellen nicht beantwortet werden kann. Hier besteht noch weiterer Forschungsbedarf, wie und ob beispielsweise mit einer systematisch gewählten Initialisierungsvariation die Anzahl der zur Überprüfung der Anforderung benötigten Anzahl reduziert wird. Eine Möglichkeit, die es zu untersuchen gilt, ist, ob mit einer Anfangsmenge (bspw. 15) an breit gestreuten Initialisierungen ähnliche Modellergebnisse hinsichtlich ihrer Struktur und/ oder der Vorhersagequalität erreicht werden. Wenn eine hohe Vorhersagequalität häufig durch eine ähnliche Modellstruktur hervorgerufen wird, ist dies ein Hinweis auf ein globales Maximum. Wenn diese Robustheitsanforderung T1 nicht erfüllt wird oder aufgrund von zur Verfügung stehenden Ressourcen nicht anzuwenden ist, lässt sich jedoch daraus noch kein sicherheitskritisches Verhalten des Modells vorhersagen. Es besteht lediglich die Möglichkeit, dass ein Modell eine geringere Generalisierbarkeit bzw. Leistungsfähigkeit besitzt als es möglich wäre. Da allerdings dieses nicht optimale Modell der Überprüfung aller anderen Anforderungen unterzogen wird, wird die Generalisierung dieses Modells analysiert und basierend hierauf ein Sicherheitskonzept ausgearbeitet, was ebenfalls die möglichen Auswirkungen des lokalen Optimums adressiert. Jedoch ist bei Verletzung der Anforderung T1 darauf zu achten, dass in den Robustheitsanforderungen, die mit neu-trainierten Modellen überprüft werden, ein ähnliches Optimum durch das neue Modell erreicht wird. Liegt dies nicht vor, resultieren die gewonnenen Erkenntnisse aus der Überprüfung der jeweiligen Robustheitsanforderung ggf. nicht aus der in den Datensatz eingebrachten Veränderung, sondern aus der veränderten Generalisierbarkeit, die durch das unterschiedliche Optimum hervorgerufen wird.

Einen zusammenfassenden Überblick über die Erkenntnisse der Robustheitsüberprüfung des Fahrstilmodells (Manöver Linksabbiegen) gibt Abbildung 6-27.

[332] Zur Erhebung einer statistisch signifikanten Aussage sind im Anwendungsfall sogar 3990 Initialisierungen notwendig.

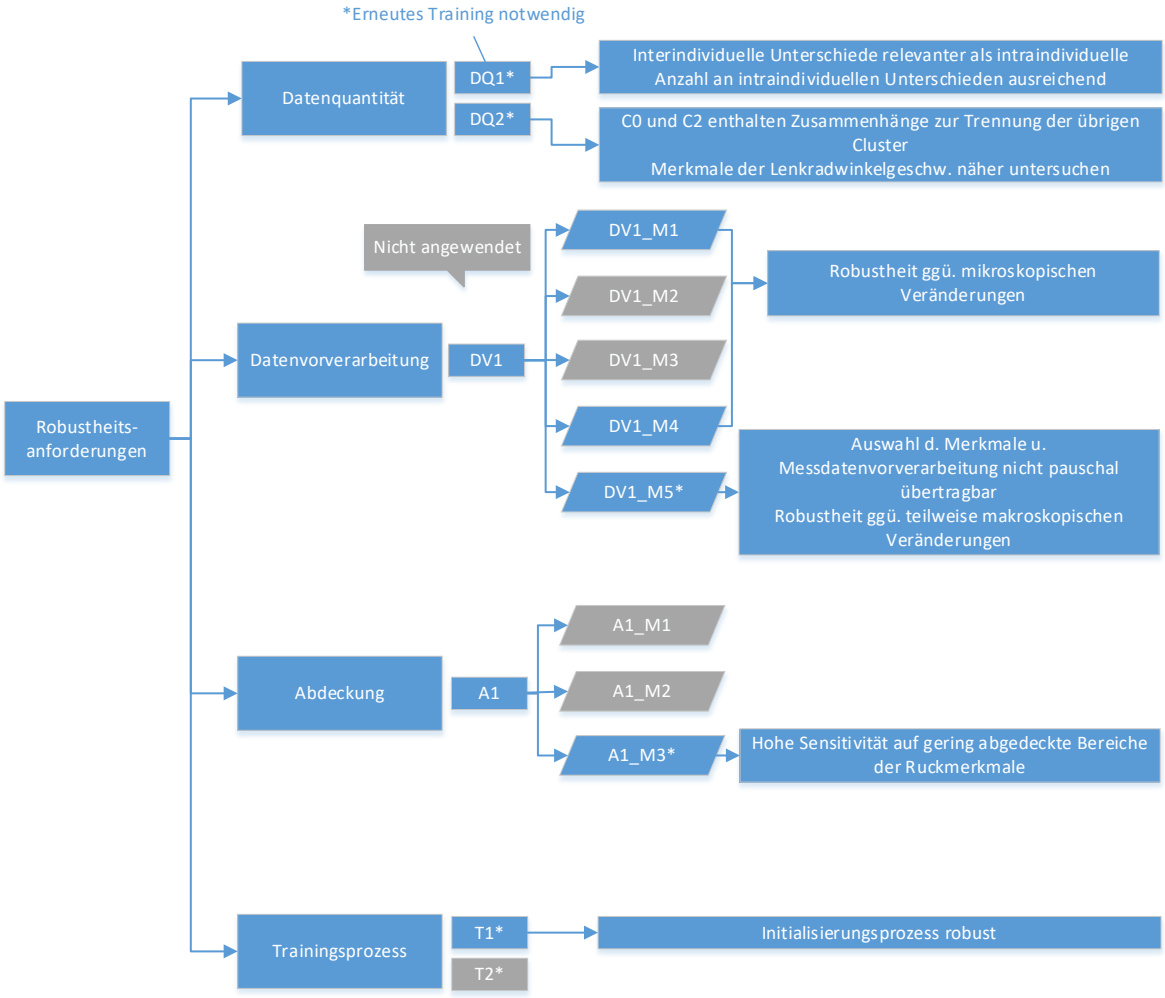


Abbildung 6-27: Überblick über erzielte Erkenntnisse der Robustheitsanforderungen

Wie erwähnt dient die Robustheitsüberprüfung unter anderem dazu, basierend auf der identifizierten Generalisierung des Modells, ein Sicherheitskonzept des Modells auszuarbeiten, damit ggf. den Auswirkungen der vorliegenden fehlenden Generalisierbarkeit entgegengewirkt werden. Im Anwendungsfall resultieren aus der Überprüfung der Robustheitsanforderungen folgende Sicherheitsanforderungen, die es umzusetzen gilt:

- Begrenzung des Betriebsbereichs auf die Bereiche, die durch Testdaten abgedeckt sind (aus Anforderung A1).
- Begrenzung der Variationsmöglichkeiten der Vorverarbeitung auf getestete Abweichungen (aus Anforderung DV1).

Die Möglichkeit der simulativen Erzeugung von Testdaten zur Überprüfung der Robustheitsanforderungen wurde bewusst nicht aufgeführt oder angewendet. Dies liegt darin begründet, dass zur Anwendung simulativ erzeugter Testdaten die Beweisführung notwendig ist, dass das zur Erzeugung genutzte Werkzeug für die Generierung von Testdaten für ML valide ist. Hierzu ist beispielsweise die Definition von Validitätskriterien notwendig, wel-

che das Simulationswerkzeug zu erfüllen hat.[333] Da die Beweisführung von Modellvalidität ein eigenes Forschungsfeld darstellt, wird sie aus der vorliegenden Betrachtung ausgeklammert. Allerdings wird darauf hingewiesen, dass die simulative Erzeugung von Testdaten nur mit der Beweisführung der Validität des genutzten Werkzeugs belastbare Erkenntnisse liefern kann.

Zusätzlich zur Überprüfung der Anwendbarkeit der Robustheitsanforderungen bei Vorliegen einer diskreten Ausgangsgröße wird in Anhang D.7 diskutiert, wie der Vergleich von mehreren kontinuierlichen Ausgangsgrößen zwischen zwei Modellen zur Überprüfung der Anforderungen möglich ist. Eine Herausforderung besteht in der Ableitung eines sinnvollen Schwellwerts, ab dem eine Abweichung der gleichen Ausgangsgrößen zwischen zwei Modellen als „Fehler" interpretiert wird. Insgesamt wird jedoch aufgrund des Vorliegens mehrerer kontinuierlicher Ausgangsgrößen keine Einschränkung der prinzipiellen Anwendbarkeit identifiziert.

Die Hypothese der Anwendbarkeit des vierten Schrittes des in Unterkapitel 5.1 abgeleiteten Ansatzes zur fehlenden Generalisierbarkeit „*Der Ansatz ist ohne Einschränkungen anwendbar*" wurde insgesamt anhand obiger Anwendung in der Überprüfung der Anforderung A1 falsifiziert. Durch die vorherrschenden Gegebenheiten wurde auf eine Methode der Anforderungsüberprüfung zurückgegriffen, die nicht direkt die Erfüllung der Anforderung des finalen Modells testet, sondern die eines Modells, welches eine hohe funktionale Ähnlichkeit zu diesem finalen Modell besitzt. Allerdings gibt es Alternativen (A1_M1 und A1_M2) zu dieser Methode A1_M3, die die Anforderung direkt adressieren, jedoch aufgrund der verfügbaren Ressourcen zur Erhebung eines neuen Datensatzes und, als Alternativlösung, zur erneuten Durchführung der Überprüfung der Robustheitsanforderungen nicht angewendet wurden. Aufgrund der bereits erwähnten Analogie in der Anwendung zwischen A1_M3 und A1_M2 wird jedoch die Anwendbarkeit lediglich auf diese zur Verfügung stehenden Ressourcen eingeschränkt. Die hieraus resultierende neue Hypothese lautet „*Der Ansatz ist allgemein anwendbar, sofern der Aufwand zur Berechnung sowie zur Erhebung neuer Datensätze getragen wird*".

Diese neue Hypothese adressiert ebenfalls die Problematik, dass der Anwendungsfall eine geringe Berechnungszeit des Trainings besitzt, was jedoch nicht bei allen Algorithmen des ML vorliegt. Robustheitsanforderungen, deren Überprüfungsmethoden auf einem häufigen Training eines Modells beruhen, sind daher aufgrund der zur Verfügung stehenden Ressourcen ggf. nicht anwendbar. Diese sind in Abbildung 6-27 mit einem Stern markiert. Alternative Vorgehensweisen, wie die Erhebung eines speziellen Testdatensatzes, werden innerhalb der Anforderungen DV1 und A1 vorgestellt. Für die Überprüfung der Anforderungen DQ1, DQ2 und T1 wurden jedoch keine Alternativen identifiziert.

[333] Vgl. Viehof, M.: Dissertation, Objektive Qualitätsbewertung von Fahrdynamiksimulationen (2018), S. 17.

6.3.3 Fazit der Anwendbarkeit

Die dritten und vierten Schritte des in Unterkapitel 5.1 entwickelten Ansatzes zur Identifikation fehlender Generalisierbarkeit wurden erfolgreich angewendet. Die Anwendung der ersten beiden Schritte dieses Ansatzes ist optional, weshalb die durchgeführte Generalisierbarkeitsüberprüfung genügt, um auf die Anwendbarkeit des Gesamtansatzes zu schließen und damit die Forschungsfrage *„Ist der Ansatz praktisch anwendbar?“* aus Unterkapitel 1.2. zu beantworten.[334] Allerdings lässt sich bedingt durch die hohe Betrachtungsebene der Ursachen fehlender Generalisierbarkeit, aus denen die ersten beiden Schritte abgeleitet wurden, fordern, dass den hierin enthaltenen Empfehlungen zur Erhöhung der Qualität im Entwicklungsprozess und direkte Überprüfungsmaßnahmen teilweise konkretere Methoden zuzuweisen sind. Hieraus leitet sich zwar keine Grenze der Anwendbarkeit des Gesamtansatzes ab, jedoch Verbesserungspotential, um eine Anwendung in den Schritten 1. und 2. zu erleichtern bzw. zu konkretisieren.

Zur Beantwortung wird die Hypothese „*Der Ansatz ist ohne Einschränkungen anwendbar*“ aufgestellt, die durch den dritten Schritt, die Überprüfung der funktionalen Anforderungen, nicht falsifiziert wird. Die Überprüfung der Robustheit des gelernten Modells falsifiziert diese Hypothese jedoch, dadurch, dass der Ansatz abhängig vom Aufwand, der zur Berechnung und Erhebung neuer Datensätze getragen wird, nur eingeschränkt anwendbar ist. Die Fragestellung nach der praktischen Anwendbarkeit wird daher wie folgt beantwortet:

> Der Ansatz ist allgemein anwendbar, sofern der Aufwand zur Berechnung sowie zur Erhebung neuer Datensätze getragen wird.

[334] Die ausführliche Begründung, warum die ersten beiden Schritte optional sind, findet sich in Unterkapitel 6.2.

7 Grenzen des Ansatzes und weitere Forschungsfragen

In Unterkapitel 6.3 wird die praktische Anwendbarkeit des Ansatzes mit Einschränkungen hinsichtlich des getragenen Aufwands, welcher zur Durchführung der Methoden zur Robustheitsüberprüfung notwendig ist, festgestellt. Die in Unterkapitel 1.2 definierte Forschungsfrage *„Welche Grenzen besitzt der Ansatz?“* zielt darüber hinaus auf die

- Grenzen des Ansatzes hinsichtlich der Erbringung eines Sicherheitsnachweises,
- Grenzen der Übertragbarkeit auf nicht-fokussierte Algorithmenkategorien,
- und Grenzen des erreichbaren Erkenntnisgewinns hinsichtlich der Generalisierbarkeit durch diesen Ansatz

ab. Dieser Frage wird im folgenden Kapitel nachgegangen und basierend hierauf weitere Forschungsfragen abgeleitet. Durch die Analyse der Grenzen des Ansatzes wird ebenfalls der Frage *„Welche weiteren Herausforderungen existieren in der Erbringung des Sicherheitsnachweises für FAS?“* aus Unterkapitel 1.2 nachgegangen.

Generell stellt der vorgestellte Ansatz lediglich einen Baustein des **Sicherheitsnachweises** für gelernte Modelle in FAS dar. Ziel des Ansatzes ist die Identifikation fehlender bzw. ausreichender Generalisierbarkeit, um hierdurch eine der Lücken zu füllen, die bestehende Sicherheitsnachweise besitzen (siehe Abschnitt 3.2.2). Es wird nicht ausgeschlossen, dass neben den zusätzlich zur fehlenden Generalisierbarkeit identifizierten ML-inhärenten Fehlerquellen, der Nutzung neuer Hardwarebausteine sowie den Gefahren hinsichtlich der veränderten System-Nutzer-Interaktion, weitere ML-inhärente Fehlerquellen bestehen.[335] Hierdurch ergibt sich die folgende offene Forschungsfrage:

„Welche weiteren ML-inhärenten Fehlerquellen und Gefahrenpotentiale existieren für Supervised- und Unsupervised-Ansätze außerhalb der Bereiche Generalisierbarkeit, genutzter Hardware und Interaktion mit dem Nutzer?“

Darüber hinaus ist der Ansatz lediglich für die bisher in FAS hauptsächlich genutzten Algorithmenkategorien des Supervised- und Unsupervised-Learning entwickelt und besitzt zusätzlich die Einschränkung der Anwendung auf offline gelernte Algorithmen. Es wird jedoch erwartet, dass auch die Nutzung von online lernenden Algorithmen bzw. Reinforcement-Learning in FAS zunimmt, um neue Funktionalitäten umzusetzen. Eine **Übertragbarkeit** des entwickelten Ansatzes ohne Änderungen und Erweiterungen auf diese Kategorien des ML ist nicht möglich, da beispielsweise die Überprüfung der funktionalen

[335] Die zugehörige Analyse zur Identifikation dieser drei Bereiche wird in Unterkapitel 3.3 durchgeführt und erhebt keinen Anspruch auf Vollständigkeit.

Anforderungen und Robustheitsanforderungen im Rahmen von online lernenden Ansätzen während des Betriebs stattfinden müsste. Durch die Unterschiede im Entwicklungsprozess von Reinforcement-Learning im Gegensatz zu Supervised- und Unsupervised-Learning werden ebenfalls andere und weitere Ursachen fehlender Generalisierbarkeit erwartet als die bisher aufgelisteten bzw. dem Ansatz zugrundeliegenden. Darüber hinaus wurde aufgrund der Fokussierung auf Supervised- und Unsupervised-Learning lediglich die bestehenden Sicherheitsnachweise dieser Kategorien untersucht, wodurch es möglich ist, dass für die Bereiche der online lernenden Algorithmen und Reinforcement-Learning andere Defizite bestehen, die nicht mit der Problematik der fehlenden Generalisierbarkeit adressiert werden. Daher werden in Ergänzung zu der oben abgeleiteten Forschungsfrage für Supervised- und Unsupervised-Ansätze eigene Forschungsfragen für Reinforcement-Learning und online gelernte Algorithmen aufgeworfen, die zusätzlich die Frage der bestehenden Sicherheitsnachweise dieser Felder beinhaltet:

„Welche Konzepte bestehen derzeit in der Erbringung des Sicherheitsnachweises für Reinforcement-Learning und online lernenden Algorithmen in sicherheitsrelevanten Systemen?“

„Welche Defizite besitzen die bestehenden Konzepte?“

Eine wesentliche Grenze des Ansatzes hinsichtlich des erzielbaren **Erkenntnisgewinns** der Generalisierbarkeit liegt in der möglichen Unvollständigkeit der Robustheitsanforderungen begründet. Die Robustheitsanforderungen werden aufgestellt, um der möglichen Unvollständigkeit der funktionalen Anforderungen zu begegnen, da bisherige Ansätze zur Lösung dieser Problematik nicht auf gelernte Modelle anwendbar sind.[336] Jedoch gehen die in dieser Arbeit aufgelisteten Robustheitsanforderungen lediglich aus der Fachliteratur hervor oder werden aus den Ursachen fehlender Generalisierbarkeit, die ebenfalls keinen Anspruch auf Vollständigkeit erheben, abgeleitet. Hierdurch ist nicht sichergestellt, dass keine weiteren möglichen Veränderungen des Entwicklungsprozesses, der Daten oder des Algorithmus bestehen, bei denen die Forderung nach Robustheit des gelernten Modells bei ausreichender Generalisierbarkeit vorliegt. Hieraus leitet sich eine weitere Forschungsfrage ab:

„(Wie) ist es möglich, eine vollständige Auflistung der Robustheitsanforderungen zu erhalten?“

Wenn es nicht möglich ist, eine vollständige Auflistung zu erreichen, ist zu analysieren, ob und mit welcher Menge an Robustheitsanforderungen eine Argumentation analog zu der des gewachsenen Testkollektivs zur Feststellung der Sicherheit von konventionellen Algorithmen bei FAS gültig ist.[337]

---

[336] Eine detaillierte Begründung ist in Unterkapitel 5.1 gegeben.

[337] Die detaillierte Erläuterung zur Argumentation, dass ein gewachsenen Testkollektiv als zusätzliche Maßnahme für konventionelle Algorithmen zum Nachweis der Sicherheit in FAS ausreicht, um möglicher Unvollständigkeit funktionaler Anforderungen zu begegnen, ist in Unterkapitel 5.1 zu finden.

Aus dieser derzeit potentiellen Unvollständigkeit leitet sich ab, dass ausreichende Generalisierbarkeit nicht über alle potentiellen Betriebsbereiche sicherstellt wird. Durch den bisherigen Ansatz ist es möglich, die Generalisierbarkeit des gelernten Modells in den überprüften Grenzen und in den überprüften Fällen festzustellen, was allerdings nicht allen möglichen Bereichen und allen möglichen Veränderungen entspricht, die dem Modell eventuell im Betrieb begegnen. Daher werden zwar ein besseres Verständnis der Funktionsweise sowie eine Einschätzung über die Robustheit des Modells durch Anwendung des Gesamtansatzes erreicht, jedoch kein Nachweis eines funktional sicheren Verhaltens im Betrieb. Eine Lösungsmöglichkeit besteht in der Eingrenzung des Betriebsbereichs auf die überprüften Bereiche.

Neben der Erweiterung der Robustheitsanforderungen lässt sich der Erkenntnisgewinn über die Generalisierbarkeit des gelernten Modells durch Erweiterung der bestehenden Fehlerbaumanalyse zur Identifikation der Ursachen fehlender Generalisierbarkeit erhöhen.[338] Wie bereits diskutiert, wird die mögliche Unvollständigkeit dieser Analyse durch die Überprüfung der Auswirkungen fehlender Generalisierbarkeit mittels funktionaler Anforderungen und Robustheitsanforderungen kompensiert, jedoch nur, solange diese Anforderungen alle Auswirkungen sicher adressieren. Da dies zum derzeitigen Entwicklungszeitpunkt des Ansatzes nicht gegeben ist (s.o.), birgt auch die Erweiterung der FTA Verbesserungspotential, wenn nicht bereits alle Ursachen identifiziert wurden.

Zusammengefasst wird die Forschungsfrage *„Welche Grenzen besitzt der Ansatz?“* daher wie folgt beantwortet:

- Der Ansatz stellt keinen eigenen Sicherheitsnachweis für gelernte Modelle in FAS dar, sondern dient lediglich der Ergänzung bestehender Nachweise.
- Der Ansatz lässt sich nicht ohne Änderungen oder Erweiterungen auf andere Kategorien des ML wie Reinforcement-Learning oder online gelernte Modelle übertragen.
- Der Erkenntnisgewinn über die Generalisierbarkeit beschränkt sich auf die überprüften Fälle und Bereiche.

Darüber hinaus werden durch obige Analyse weitere notwendige Betrachtungen zum Nachweis der Sicherheit von ML für FAS identifiziert, wodurch die Forschungsfrage *„Welche weiteren Herausforderungen existieren in der Erbringung des Sicherheitsnachweises für FAS?“* beantwortet wird:

[338] Siehe Unterkapitel 4.1.

Für die Anwendung von offline gelernten Supervised- und Unsupervised-Learning-Ansätzen in FAS bestehen zwei Herausforderungen in der Erbringung eines Sicherheitsnachweises: Als ersten Schritt ist die vollständige Beweisführung zu erbringen, dass alle ML-inhärenten Fehlerquellen hinreichend identifiziert werden. Wenn diese Herausforderung bewältigt wurde, ist der Nachweis der praktischen Anwendbarkeit der bestehenden Sicherheitsnachweise zu führen.

Für die Erbringung des Sicherheitsnachweises von online gelernten Algorithmen sowie Reinforcement-Learning in FAS bestehen aufgrund der fehlenden Übertragbarkeit der in der vorliegenden Arbeit identifizierten Methoden aus dem Supervised- und Unsupervised-Bereich weitere Herausforderungen. Zunächst ist analog zum Vorgehen der vorliegenden Arbeit die Analyse der bestehenden Sicherheitsnachweise für diese Algorithmenarten hinsichtlich vorliegender Defizite notwendig, um anschließend den möglichen Defiziten strukturiert zu begegnen. Darüber hinaus ist der Nachweis der praktischen Anwendbarkeit zu führen.

8 Zusammenfassung und Ausblick

Die Analyse der derzeitigen Methoden zur Erbringung des Sicherheitsnachweises für konventionell programmierte Modelle in FAS zeigt, dass diese Methoden nicht generell auf gelernte Modelle übertragbar sind. Findet eine Begrenzung des Arbeitsbereichs des gelernten Modells statt, ist es möglich einen testfallbasierten Nachweis der Sicherheit zu führen. Hierdurch wird der Funktionsumfang der Modelle im Betrieb stark eingeschränkt. Eine teilweise analytische Beweisführung wird durch die Nutzung von (teilweise) interpretierbaren Modellen ermöglicht. Die Bereiche dieser Modelle, deren Sicherheit nicht analytisch beweisbar ist, sind mit Testfällen zu adressieren. Die hierzu benötigte Testfallanzahl ist gegenüber einem vollständigen Beweis der Sicherheit bei Vorliegen einer Black-Box je nach Stärke der Interpretierbarkeit vermindert. Jedoch grenzt dieses Vorgehen die Auswahl der zur Verfügung stehenden Lernalgorithmen auf Modelle mit beschränkter Leistungsfähigkeit ein. Die Alternative zu diesen Methoden besteht in der Nutzung strukturierter Sicherheitsnachweise, die bereits für den Einsatz von ML in sicherheitsrelevanten Systemen entwickelt wurden. Diese Nachweise besitzen allerdings den Nachteil, dass sie auf der Annahme basieren, dass alle ML-inhärenten Fehler identifiziert werden. Die Annahme ist jedoch durch den derzeitigen Stand der Forschung nicht erfüllt, weshalb nach Möglichkeiten zur vollständigen Identifikation dieser Fehler geforscht wird. Einer dieser ML-inhärenten Fehler begründet sich in möglicher unzureichender Generalisierbarkeit der erlernten Zusammenhänge, worauf sich im weiteren Verlauf der vorliegenden Arbeit fokussiert wird. Abbildung 8-1 stellt diese Zusammenhänge übersichtlich dar.

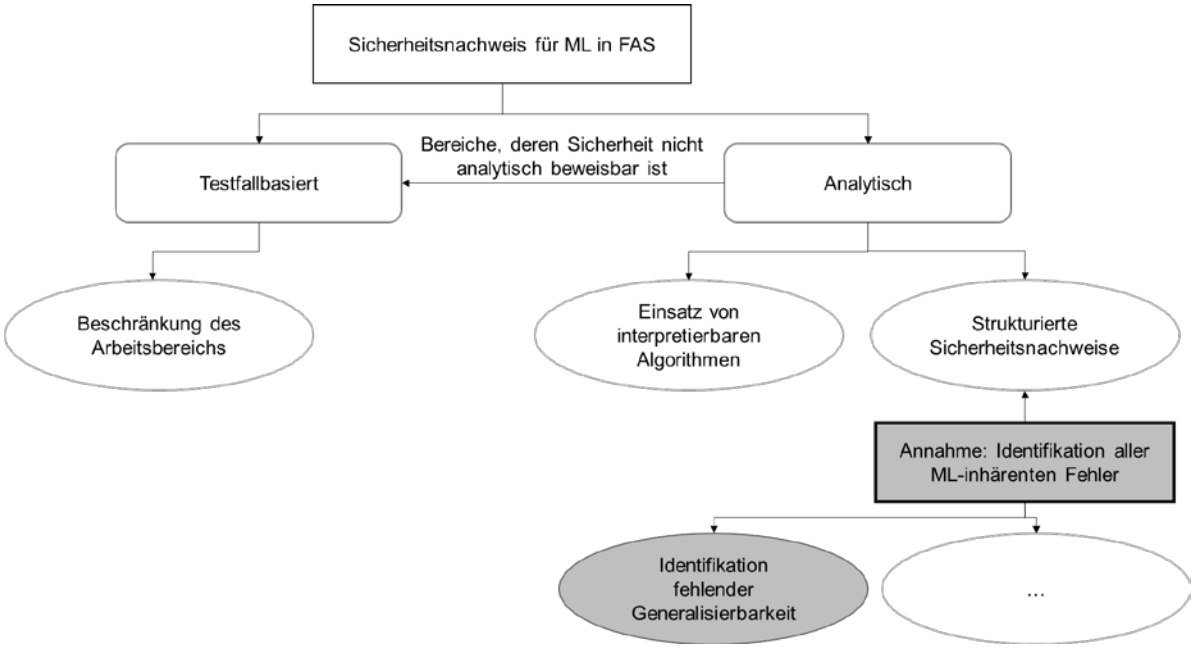


Abbildung 8-1: Fehlende Generalisierbarkeit im Gesamtkontext

Der entwickelte Ansatz zur Identifikation fehlender Generalisierbarkeit bildet daher einen wichtigen Baustein zur Ergänzung bestehender strukturierter Sicherheitsnachweise. Durch die Überprüfung der Ursachen und Auswirkungen fehlender Generalisierbarkeit in den einzelnen Schritten des Ansatzes wird fehlendende Generalisierbarkeit systematisch untersucht und reduziert. Die ersten beiden Schritte sind dabei während der Entwicklung zu berücksichtigen und ermöglichen frühzeitig die Verbesserung des Modells hinsichtlich der vorliegenden Generalisierbarkeit durch die Vermeidung der Ursachen. Die Schritte drei und vier sind nach der Entwicklung anzuwenden, wobei iterative Verbesserungen des Modells auf Basis der gewonnenen Erkenntnisse nicht auszuschließen sind. Durch die Überprüfung der funktionalen Anforderungen im dritten Schritt ist es bei Modellen des Unsupervised-Learning möglich, funktional falsches Verhalten zu identifizieren, auch wenn keine Ground-Truth zur Verfügung steht. Doch auch bei Supervised-Learning besitzt dieser Schritt Relevanz, da hierdurch beispielsweise überprüft wird, ob mikroskopische anstelle makroskopischer Zusammenhänge zur Vorhersage genutzt werden. Der vierte Schritt untersucht durch die Überprüfung der allgemeingültigen Robustheitsanforderungen die Stabilität bzw. Sensitivität des gelernten Modells auf Veränderungen. Bei Annahme des Vorliegens von Generalisierbarkeit wird erwartet, dass diese Veränderungen keine Auswirkung des funktionalen Verhaltens des Modells nach sich ziehen. Jede Robustheitsanforderung adressiert daher einen anderen Bereich der Generalisierbarkeit, wodurch umfassende Erkenntnisse über diese gewonnen werden.

Die prototypische Anwendung zeigt, dass der Ansatz zumindest auf einfache Clustering-Algorithmen anwendbar ist. Durch die Ähnlichkeit der Modelle wird eine Anwendbarkeit auf einfache Klassifikationsalgorithmen ebenfalls erwartet. Der Anwendbarkeit werden je nachdem, ob der Aufwand getragen wird, Grenzen durch die benötigten bzw. zur Verfügung stehenden Ressourcen, wie Zeit und Rechenleistung, gesetzt, da in zwei der fünf Robustheitsanforderungen ein häufiges Training des gleichen Algorithmus unter leicht veränderten Bedingungen zur Überprüfung der Anforderungen erforderlich ist. Darüber hinaus wird die Definition von funktionalen Anforderungen bei breitem Funktionsumfang und komplexen Algorithmen zur Herausforderung. Hierzu werden einige Alternativen zur herkömmlichen Anforderungsdefinition vorgeschlagen, die allerdings im Rahmen der prototypischen Anwendung durch die Nutzung eines Anwendungsfalls aus einem bereits gut erforschten Bereich nicht notwendig sind, weshalb sie nicht angewendet wurden. Die Grenzen des Ansatzes bzw. der durch den Ansatz möglichen Erkenntnisse hinsichtlich der Generalisierbarkeit liegt vor allem in der nicht auszuschließenden Unvollständigkeit der Ursachen bzw. der Robustheitsanforderungen begründet. Hierdurch ist es nicht möglich, die Generalisierbarkeit vollständig zu überprüfen. Mit einer Erweiterung der Fehlerbaumanalyse zur Erfassung weiterer Ursachen sowie der Definition von weiteren Robustheitsanforderungen wird noch ein hohes Potential für die durch den Ansatz identifizierbare Generalisierbarkeit erwartet.

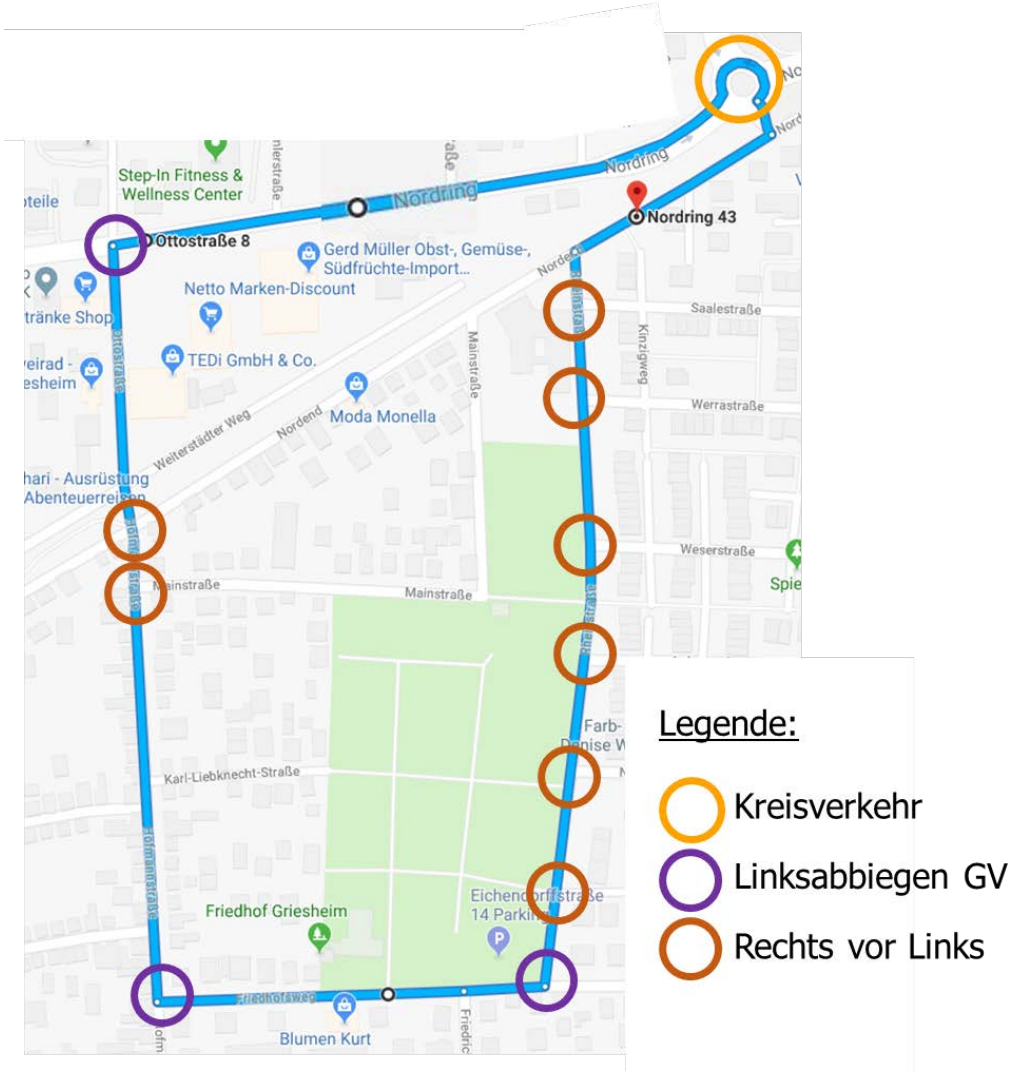
Neben den bereits in Kapitel 7 definierten Forschungsfragen und dem in Abschnitt 6.3.3 identifizierten Verbesserungspotential zur konkreteren Ausformulierung der in den ersten

beiden Schritten des Gesamtansatzes anzuwendenden Methoden bildet eine generelle Erkenntnis der vorliegenden Arbeit, dass ein wichtiges Forschungsfeld in den benötigten Eigenschaften eines Datensatzes für den Einsatz in ML liegt, um eine ausreichende Generalisierbarkeit zu gewährleisten. Neben der grundsätzlichen Frage, wie viele Daten für die Bewältigung einer bestimmten Aufgabe mit ML mit ausreichender Generalisierbarkeit notwendig sind, ist zu analysieren, welche Struktur der Daten zur Erreichung dieser Generalisierbarkeit führen und welche Einflussparameter hierfür zu variieren sind. Im Rahmen der vorliegenden Arbeit wird bspw. festgestellt, dass interindividuelle Zusammenhänge zur Detektion von Fahrstilen eine höhere Relevanz zur Generierung einer ausreichenden Generalisierbarkeit besitzen als intraindividuelle Zusammenhänge. Allerdings liegt ein solches Wissen vor Erhebung der Daten und Nutzung der Daten zumeist nicht vor, wodurch eine viel höhere Menge an Daten zu erfassen ist als zwingend notwendig. Ein Ansatz zur Bestimmung einer minimal notwendigen Datenmenge besteht in der Nutzung von Methoden der Statistik. Diese werden beispielsweise zur Abschätzung, wie viele Probanden für allgemeine Erkenntnisse (ohne die Nutzung von ML) aus Fahrversuchen benötigt werden, angewendet.[339] Wenn sich durch diese berechnete Datenmenge allgemeine Erkenntnisse durch Analyse der Daten gewinnen lassen, liegt die Vermutung nahe, dass aus einer solchen statistisch berechneten Datenmenge ebenfalls eine ausreichende Generalisierbarkeit für gelernte Modelle resultiert, solange die Auswahl der Eingangsgrößen für die Aufgabenerfüllung korrekt gewählt ist. Ob und unter welchen Voraussetzungen, wie beispielsweise einem maximal vorhandenem Messrauschen, sich diese Vermutung bestätigen lässt, gilt es neben der Analyse weiterer Ansätze zur Bestimmung ausreichender Datenqualität und -quantität in weiterer Forschung zu untersuchen.

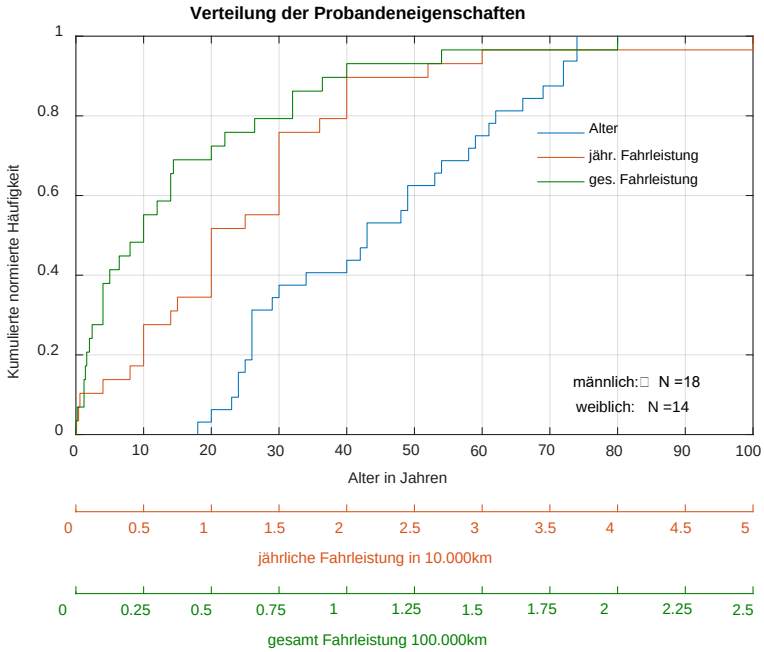
[339] Vgl. Bubb, H.: Wie viele Probanden braucht man für allgemeine Erkenntnisse aus Fahrversuchen? (2003).

A Hintergrund zum Anwendungsfall

A.1 Probandenversuch



Anhang 8-1: Streckenführung Probandenstudie in 64347 Griesheim



Anhang 8-2: Verteilung der Probandeneigenschaften Alter, jährliche Fahrleistung und Gesamtfahrleistung

A.2 Sicherheitsanalyse Use-Case Linksabbiegen

A.2.1 Gefahrenanalyse

Nr.	Funktion	Fehlfunktion	Betriebszustand	Fahrsituation	Gefahrenszenario	Gefährdung	Kommentar
1	**Empfehlung „rot“ ausgeben**	zu konservativ	-	-	Keine Gefährdung	-	
2	siehe Nr. 1	zu sportlich	-	-	Keine Gefährdung	-	
3	siehe Nr. 1	Konstanz der Empfehlung zu gering	aus dem Stand	Linksabbiegen bei Gegenverkehr	Fahrer fährt bei vermeintlich passender Lücke (grün) los, welche sich anschließend als zu kurz (rot) herausstellt. Der Fahrer beachtet die Anzeige im Abbiegevorgang allerdings nichtmehr, wodurch es zur Kollision kommt.	Unfall mit Gegenverkehr	identisch mit Fehlfunktion der Funktion "Empfehlung "grün" ausgeben"
4	siehe Nr. 1	siehe Nr. 3	siehe Nr. 3	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
5	siehe Nr. 1	siehe Nr. 3	aus der Schrittgeschwindigkeit	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 3	Unfall mit Gegenverkehr	identisch mit Fehlfunktion der Funktion "Empfehlung "grün" ausgeben"
6	siehe Nr. 1	siehe Nr. 3	siehe Nr. 5	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
7	siehe Nr. 1	siehe Nr. 3	während der Konstantfahrt (v> Schrittgeschwindigkeit)	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 3	Unfall mit Gegenverkehr	identisch mit Fehlfunktion der Funktion "Empfehlung "grün" ausgeben"
8	siehe Nr. 1	siehe Nr. 3	siehe Nr. 7	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	

9	siehe Nr. 1	siehe Nr. 3	während einer (kleinen) positiven Beschleunigung	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 3	Unfall mit Gegenverkehr	identisch mit Fehlfunktion der Funktion "Empfehlung "grün" ausgeben"
10	siehe Nr. 1	siehe Nr. 3	siehe Nr. 9	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
11	siehe Nr. 1	siehe Nr. 3	während einer negativen Beschleunigung	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 3	Unfall mit Gegenverkehr	identisch mit Fehlfunktion der Funktion "Empfehlung "grün" ausgeben"
12	siehe Nr. 1	siehe Nr. 3	siehe Nr. 11	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
13	**Empfehlung „grün“ ausgeben**	zu konservativ	-	-	Keine Gefährdung	-	
14	siehe Nr. 13	zu sportlich	aus dem Stand	Linksabbiegen bei Gegenverkehr	Fahrer verlässt sich auf die Empfehlung und nimmt eine für ihn zu kleine Lücke.	Unfall mit Gegenverkehr	
15	siehe Nr. 13	siehe Nr. 14	siehe Nr. 14	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
16	siehe Nr. 13	siehe Nr. 14	aus der Schrittgeschwindigkeit	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 14	Unfall mit Gegenverkehr	
17	siehe Nr. 13	siehe Nr. 14	siehe Nr. 16	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
18	siehe Nr. 13	siehe Nr. 14	während der Konstantfahrt (v> Schrittgeschwindigkeit)	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 14	Unfall mit Gegenverkehr	
19	siehe Nr. 13	siehe Nr. 14	siehe Nr. 18	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
20	siehe Nr. 13	siehe Nr. 14	während einer (kleinen) positiven Beschleunigung	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 14	Unfall mit Gegenverkehr	
21	siehe Nr. 13	siehe Nr. 14	siehe Nr. 20	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	

22	siehe Nr. 13	siehe Nr. 14	während einer negativen Beschleunigung	Linksabbiegen bei Gegenverkehr	siehe Gefahrenszenario Nr. 14	Unfall mit Gegenverkehr	
23	siehe Nr. 13	siehe Nr. 14	siehe Nr. 22	Linksabbiegen ohne Gegenverkehr	Keine Gefährdung	-	
24	siehe Nr. 13	Konstanz der Empfehlung zu gering	-	-	-	-	identisch mit Fehlfunktion der Funktion "Empfehlung "rot" ausgeben"
25	**Keine Empfehlung ausgeben, da Lernmodus (fehlende Konfidenz)**	zu konservativ	-	-	Keine Gefährdung	-	
26	siehe Nr. 25	zu sportlich	-	-	Keine Gefährdung	-	
27	siehe Nr. 25	Konstanz der Empfehlung zu gering	-	-	Keine Gefährdung	-	

Anhang 8-3: Gefahrenanalyse des Use-Cases Linksabbiegen

A.2.2 FMEA

Die Berechnung des Automotive-Integrity-Levels (ASIL) für den Use-Case Linksabbiegen erfolgt nach den Vorgaben der ISO 26262:2018 mit der Durchführung einer FMEA und ist in Anhang 8-4 dargestellt. Das ASIL berechnet sich aus drei Faktoren: Severity, Exposure und Controllability, die es für jede Gefährdung separat zu bestimmen gilt. Das ASIL bezeichnet eine von vier Risikostufen aus denen sich Sicherheitsanforderungen für die Bausteine des Systems ableiten, für die das ASIL gilt. ASIL A bezeichnet die niedrigste Stufe mit den geringsten hieraus folgen-

den Sicherheitsanforderungen und ASIL D die höchste Stufe. Die Bezeichnung „QM“ bedeutet, dass das aus der Gefährdung resultierende Risiko so gering ist, dass ein Qualitätsmanagement ausreicht und dass die Funktion nicht zwangsweise nach ISO 26262 zu entwickeln ist.340

Die Severity stellt die Schwere der Verletzung dar und wird von S0 bis S3 angegeben. S0 entspricht keinen Verletzungen, S1 bedeutet leichte und gemäßigte Verletzungen, S2 sind schwere und lebensbedrohliche Verletzungen (Überleben möglich) und S3 umfasst lebensbedrohliche Verletzungen (Überleben ungewiss) sowie tödliche Verletzungen. Die in der Norm aufgelisteten Beispiele für die einzelnen Verletzungsklassen wurden zur Kategorisierung der Schwere genutzt. Leichte und gemäßigte Verletzungen resultieren beispielsweise aus einem seitlichen Zusammenstoß eines Fahrzeugs mit einem stehenden Objekt (z.B. ein Baum) mit sehr geringer Geschwindigkeit.341a Exposure bezeichnet die Aufenthaltswahrscheinlichkeit des Fahrers in einer Situation, in der die betreffende Fehlfunktion möglich ist. E4 stellt Situationen mit einer sehr hohen Wahrscheinlichkeit dar, die fast jede Fahrt einmal auftreten.341b Hier ist der Use-Case Linksabbiegen einzuordnen, da man gemittelt über alle Fahrer anzunehmen hat, dass mindestens einmal pro Fahrt bei Vorfahrt des Gegenverkehrs links abgebogen wird. Der Faktor Controllability schätzt die Wahrscheinlichkeit, dass ein repräsentativer Fahrer oder andere beteiligte Personen in der Lage sind die Situation so zu beeinflussen, dass die Gefährdung gemindert wird. Diese Kontrollierbarkeit wird von C0 bis C3 bemessen, wobei C0 aussagt, dass eine Situation allgemein kontrollierbar ist und C3, dass eine Situation schwierig oder unmöglich zu kontrollieren ist. 341c Die Zuordnung zu C1 in allen Gefährdungssituationen rührt daher, dass die Situation „einfach“ kontrollierbar ist, indem der Fahrer oder der Fahrer des entgegenkommenden Fahrzeugs bremst, falls die Zeitlücke nichtmehr für ein kollisionsfreies Abbiegen genügt. Da das System im städtischen Bereich mit durchschnittlichen Geschwindigkeiten von 50 km/h eingesetzt wird und lediglich als „Empfehlungssystem“ angepriesen wird, ist diese Einschätzung realistisch. Die aus den einzelnen Faktoren resultierende ASIL bzw. QM-Klassifizierung ist Anhang 8-5 zu entnehmen.

---

[340] Vgl. ISO: ISO 26262:2018. Road vehicles: Functional safety (2018), Teil 1, S. 2.

[341] Vgl. ISO: ISO 26262:2018. Road vehicles: Functional safety (2018), a: Teil 3, S. 21; b: Teil 3, S. 23; c: Teil 3, S. 27.

Nr.	Severity	Kommentar(S)	Exposure	Kommentar(E)	Controllability	Kommentar(C)	ASIL/QM
3	1	Seitlicher Zusammenstoß mit sehr geringer Geschwindigkeit	4	>10% Aufenthaltsdauer während der Betriebszeit	C1	Fahrer erkennt die für ihn zu kleine Lücke und bremst vor dem Abbiegevorgang.	**QM**
5	1	Seitlicher Zusammenstoß mit sehr geringer Geschwindigkeit	4	s.o.	C1	s.o.	**QM**
7	3	Seitlicher Zusammenstoß mit mittlerer Geschwindigkeit	4	s.o.	C1	s.o.	**B**
9	3	Seitlicher Zusammenstoß mit mittlerer Geschwindigkeit	4	s.o.	C1	s.o.	**B**
11	2	Seitlicher Zusammenstoß mit geringer Geschwindigkeit	4	s.o.	C1	s.o.	**A**
14	1	Seitlicher Zusammenstoß mit sehr geringer Geschwindigkeit	4	s.o.	C1	s.o.	**QM**
16	1	Seitlicher Zusammenstoß mit sehr geringer Geschwindigkeit	4	s.o.	C1	s.o.	**QM**
18	3	Seitlicher Zusammenstoß mit mittlerer Geschwindigkeit	4	s.o.	C1	s.o.	**B**
20	3	Seitlicher Zusammenstoß mit mittlerer Geschwindigkeit	4	s.o.	C1	s.o.	**B**
22	2	Seitlicher Zusammenstoß mit geringer Geschwindigkeit	4	s.o.	C1	s.o.	**A**

Anhang 8-4: Fehlermöglichkeits- und -einflussanalyse Use-Case Linksabbiegen

Severity class	Exposure class	Controllability class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A[a]
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

[a] See 6.4.3.11.

Anhang 8-5: Bestimmung des ASIL[342]

[342] ISO: ISO 26262:2018. Road vehicles: Functional safety (2018), Teil 3, S. 10.

A.3 Sicherheitskonzept Use-Case Linksabbiegen

Das Sicherheitskonzept besteht aus Sicherheitsanforderungen, die benötigt werden, um die Sicherheitsziele zu erreichen. Zur Ableitung von Sicherheitszielen werden die Fehlfunktionen weiter betrachtet, die mindestens ASIL A aus der FMEA erhalten haben. Fehlfunktionen, die lediglich zu QM führen, werden nicht innerhalb der ISO 26262 betrachtet, sondern sind mit regulären Methoden des Qualitätsmanagements handhabbar.[343] Werden einer Fehlfunktion mehrere unterschiedliche ASIL basierend auf unterschiedlichen Betriebszuständen zugrunde gelegt, wird das höchst aufgetretene ASIL dieser Fehlfunktion zugeordnet.[344] Die aus der FMEA des Use-Cases Linksabbiegen hervorgegangenen Sicherheitsziele beziehen sich daher auf die Fehlfunktionen „Konstanz der Empfehlung zu gering“ und „zu kleiner Schwellwert für den Fahrer“. Beide Fehlfunktionen besitzen ASIL B. Hieraus leiten sich folgende Sicherheitsziele ab:

- Sicherheitsziel 1: Die Empfehlung hat in einem gewissen Zeitraum konstant zu bleiben. Der Wechsel der Empfehlung einer Lücke ist dem Fahrer deutlich zu machen.
- Sicherheitsziel 2: Der Fahrer darf keinen zu kleinen Lückenschwellwert erhalten.

Um diese Sicherheitsziele zu erreichen, ist die Erstellung eines Sicherheitskonzepts und hieraus abgeleitet die Definition von Sicherheitsanforderungen notwendig. Das erste Sicherheitsziel bezieht sich nicht auf das gelernte Modell der Fahrstildetektion, da es auch bei korrekter Funktionsweise des Modells auftritt. Hierdurch ist es zwar für die generelle Sicherheit des City Assistant Systems zu erfüllen, jedoch zur Bestimmung der Sicherheitsrelevanz des gelernten Modells irrelevant, weshalb die Erfüllung dieses Sicherheitsziels an dieser Stelle nicht weiter ausgeführt wird.

Das zweite Sicherheitsziel bezieht sich direkt auf eine korrekte Funktionalität des gelernten Modells, da dies für die Zuordnung des Fahrers zu Fahrstilen genutzt wird aus welchen sich der individuelle Lückenschwellwert ableitet. Es ist zu prüfen, ob andere Maßnahmen existieren, die dazu führen, dass trotz inkorrekter Funktionalität, d.h. Zuordnung des Fahrers zu einem Cluster mit einer höheren Risikoakzeptanz als er selbst, das Sicherheitsziel erreicht wird. Eine Möglichkeit besteht darin, die Cluster nicht mit dem Lückenakzeptanzschwellwert, der für dieses Cluster aus den Daten bestimmt wurde, zu verknüpfen, sondern jeweils mit dem des Clusters, dessen Schwellwert eine Stufe höher ist. Praktisch entspricht dies in der aktuellen Auslegung des City Assistant Systems einem Schwellwert-Offset von 0,5 Sekunden. Allerdings verbleibt hier die Problematik, dass lediglich die fälschliche Eingruppierung des Fahrers in das Nachbarcluster verhindert wird und nicht eine falsche Ein-

[343] Vgl. ISO: ISO 26262:2018. Road vehicles: Functional safety (2018), Teil 1, S. 2.

[344] Vgl. Gebhardt, V.: Funktionale Sicherheit nach ISO 26262 (2013).

gruppierung in Cluster, deren Schwellwert sich noch weiter entfernt befindet. Zusätzlich wurde im Rahmen des Sicherheitskonzepts des ersten Sicherheitsziels bereits ein Zeitpuffer auf den Schwellwert von 0,5 Sekunden festgelegt, um den Kollisionsbereich in der Kreuzung trotz eingeleiteter Bremsung des Fahrers aufgrund der Warnung vor einer Lücke zu passieren. Durch einen zusätzlichen Offset von 0,5 Sekunden ist der Lückenschwellwert dann um 1,0 Sekunden erhöht, wodurch der wahre, durch Daten erhobene, Schwellwert des risikobewusstesten Clusters dem des risikounbewusstesten Clusters entspricht. Die Akzeptanz des Systems bei einer so konservativen Auslegung ist in Frage zu stellen, da risikounbewusste Fahrer häufig Zeitlücken im Gegenverkehr angeboten bekommen würden, die das System nicht empfiehlt, sie aber dennoch gerne nehmen würden. Daher wird die Möglichkeit des Offsets ausgeschlossen. Eine weitere Lösungsmöglichkeit besteht in der Entwicklung eines redundanten, konventionell programmierten Modells zur Fahrstildetektion, auf welches bei geringer Konfidenz des gelernten Modells vertraut wird. Die Bestimmung der Konfidenz der Clusterzuordnung erfolgt beispielsweise über die Angabe der Distanz zu den einzelnen Clusterschwerpunkten. Es ist von einer niedrigen Zuverlässigkeit der Aussage des gelernten Modells auszugehen, wenn ein Datenpunkt eine hohe Distanz zu allen Clusterschwerpunkten aufweist, was durch den verwendeten k-Means Algorithmus gleichbedeutend ist mit einem Datenpunkt, der eine große Distanz zu den Datenpunkten des Trainingsdatensatzes aufweist und daher als „Ausreißer“ zu bewerten ist. Hierbei gilt die Annahme, dass der Trainingsdatensatz repräsentativ hinsichtlich des Gesamtfahrerkollektivs ist. Auf ein konventionell erstelltes Modell ist jedoch in diesem Fall ebenfalls nicht zurückzugreifen, da dieses anhand der Studiendaten bzw. Trainingsdaten parametriert wird und die Annahmen in gering abgedeckte Bereiche extrapoliert, wofür, genauso wie im Fall eines gelernten Modells, keine Begründung der Korrektheit möglich ist. Daher scheidet auch diese Möglichkeit als Alternative aus. Es ist daher die korrekte Einordnung des Fahrers zu beweisen.

A.4 Prozesskette der Datenverarbeitung

Die zum Training des Fahrstilmodells genutzten Daten wurden durch Fahrten mit einem Versuchsträger auf öffentlichen Straßen erhoben (siehe Abschnitt 6.1.3). Die zum Training genutzten Datenkanäle werden aus dem Fahrzeug-CAN ausgelesen und mit einem gemeinsamen Zeitstempel synchronisiert. Die Zeitspanne zwei Zeitstempeln beträgt ca. 20 ms. Es existieren leichte Schwankungen in dieser Zeitspanne, da diese Synchronisierung in Echtzeit vorgenommen wird. Zur Festlegung der Werte der einzelnen Datenkanäle pro Zeitstempel wird der letzte Wert des Datenkanals genutzt, der vorlag. Die im Modell verwendeten Datenkanäle sind:

- Geschwindigkeit
- Längsbeschleunigung

- Querbeschleunigung
- Lenkradwinkelgeschwindigkeit

Nach jeder abgeschlossenen Runde der Versuchsstrecke wird die Messung neu gestartet, wodurch pro Versuchsrunde ein eigener Teildatensatz existiert.

Zur Anpassung dieser Teildatensätze für das gelernte Manövermodell des Linksabbiegens, wie es in Unterkapitel 6.2 genutzt, wird aus den Zeitreihen der Signale das Manöver Linksabbiegen extrahiert. Das Manöver wird auch im Fahrzeug retrospektiv analysiert, d.h. erst nach der Manöverdurchführung wird ein Linksabbiegen im Stillstand identifiziert. Generell wird jedes Fahrmanöver welches einen Abstand von 50 m zum Mittelpunkt der Kreuzung unterschreitet als potentielles Linksabbiegen aus dem Stillstand näher betrachtet. Das Ende des Betrachtungszeitraums stellt der Datenpunkt dar, bei welchem der Abstand zum Mittelpunkt der Kreuzung die 50 m wieder überschreitet. Der Abstand zum Mittelpunkt der Kreuzung wird durch das GPS-Signal des Ego-Fahrzeugs bestimmt.

Da die Datenerhebung strukturiert stattfand, bildet im vorliegenden Fall jedes Fahrmanöver an der betreffenden Kreuzung ein Linksabbiegen. Hierdurch ist nur noch zu unterschieden, ob das Abbiegen aus dem Stillstand oder aus der Fahrt erfolgte. Als Unterscheidungskriterium wird die minimale Geschwindigkeit im Manöververlauf genutzt. Liegt diese unter 2,5 m/s wird ein Linksabbiegen aus dem Stillstand klassifiziert.

Im nächsten Schritt wird der exakte Startpunkt des Linksabbiegens aus dem Stillstand definiert, da der bisherige Betrachtungsraum des Fahrmanövers von 50 m vor der Kreuzung ebenfalls die Annäherung an die Kreuzung enthält, welche jedoch ein eigenes Manöver darstellt. Folgende Kriterien legen den Startpunkt fest:

- Der Startdatenpunkt liegt vor dem in der Betrachtungszeitraum auftretenden maximalen Lenkradwinkel
- Der Startdatenpunkt ist der letzte Datenpunkt bei dem die minimal aufgetretene Geschwindigkeit während Betrachtungszeitraums plus 0,1 m/s kleiner ist als der aktuelle Datenpunkt.

Der Endpunkt des Linksabbiegemanövers wird bei 50 m zum Kreuzungsmittelpunkt belassen. Durch den gesamten Vorgang werden die Zeitreihen der Signale auf das Linksabbiegemanöver gekürzt. Es findet für das Training des gelernten Modells keine weitere Vorverarbeitung der Signale, wie eine Filterung, statt. Daher werden aus diesen Manöver-Zeitreihen die zum Training genutzten statistischen Signale abgeleitet.

B Parameter K-Means

Zur Implementierung des K-Means Algorithmus wurde auf die Programmiersprache Python zurückgegriffen. Dabei enthält die scikit-learn Bibliothek eine vorgefertigte Funktionalität zum Training eines K-Means Modells.[345] Für die Implementierung sind die in der ersten Spalte in Anhang 8-6 gelisteten Parameter zu definieren. Alle Parameter besitzen bereits einen Standardwert. Wird dieser beibehalten, ist das durch eine graue Schriftfarbe in der dritten Spalte gekennzeichnet. Bei Änderung des Standardwerts wird schwarze Schriftfarbe genutzt. Weiterführende Beschreibungen der Parameter sowie mögliche Werte sind unter scikit-learn.org[346] [347] dokumentiert.

Parameter	Beschreibung	Wert
$n_{clusters}$	Anzahl an gewünschten Clustern	3
init	Methode der Initialisierung „k-means ++“: Methode zur intelligenten Auswahl der Clusterschwerpunkte, um die Konvergenz zu beschleunigen.[348]	k-means++
n_{init}	Anzahl an unterschiedlichen zufälligen Initialisierungen der Clusterschwerpunkte	70
max_{iter}	Maximale Anzahl an Iterationsschritten zur Modellfindung	300
tol	Schwellwert der Clusteränderung, ab welchem das Training abgebrochen wird, da Konvergenz prädiziert wird.	0,0001
$precompute_{distances}$	Vorherige Berechnung der Distanzen, wodurch das Modell schneller trainiert wird. Die Methode erfordert jedoch mehr Speicherplatz. „auto“: Wenn die Multiplikation aus der Anzahl an Datenpunkten mit der Anzahl an Clustern größer als 12 Millionen ist, wird die vorherige Berechnung nicht	auto

[345] Vgl. scikit-learn: sklearn.cluster.KMeans (2019).

[346] scikit-learn: sklearn.cluster.KMeans (2019).

[347] scikit-learn: 2.3. Clustering — scikit-learn 0.20.3 documentation (2019).

[348] Siehe Abschnitt 6.2.3.4.

	durchgeführt, andernfalls schon.	
verbose	Programmmodus, in welchem die Zwischenschritte der Ausführung ausführlich dokumentiert werden.	0
***random*$_{\text{state}}$**	Modus, bei welchem die „deterministische“ Auswahl der zufälligen Initialisierungen der Clusterschwerpunkte möglich ist	None
***copy*$_{\text{x}}$**	Auswahl der Speicherplatzanordnung „True“: Originaldaten werden nicht geändert	True
***n*$_{\text{jobs}}$**	Anzahl der für die Berechnung zu verwendenden Prozessoren. „-1“: Nutzung aller zur Verfügung stehenden Prozessoren	-1
algorithm	Berechnungsmethode des Algorithmus. Es stehen zwei Varianten zur Verfügung: Die „volle“ traditionelle Weise zur Berechnung des K-Means oder die „elkan“-Variante, die durch die Verwendung der Dreiecksungleichheit effizienter arbeitet, jedoch derzeit nicht für gering abgedeckte Bereiche des Datensatzes nutzbar ist. „auto“: Nutzung der klassischen Berechnung für wenig abgedeckte Bereiche des Datensatzes und Nutzung der „elkan“-Variante für dicht abgedeckte Datenbereiche.	auto

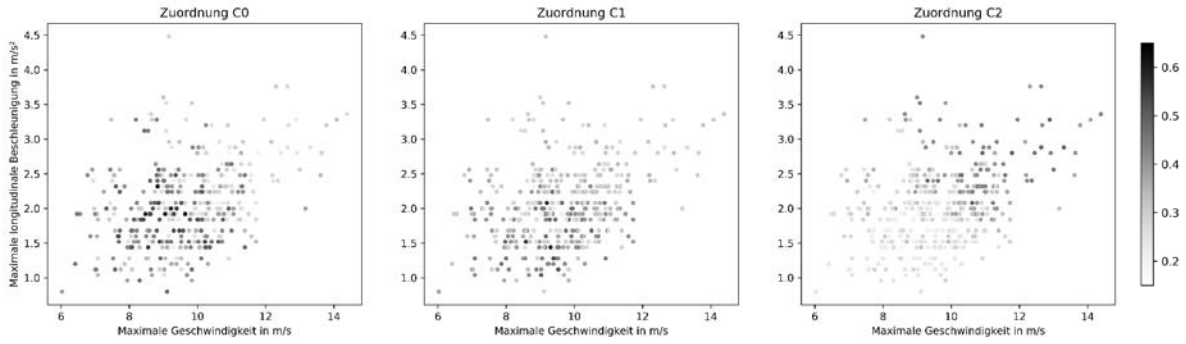
Anhang 8-6: Parametrierung K-Means[349]

[349] Vgl. scikit-learn: sklearn.cluster.KMeans (2019). für Parameter und Beschreibungen.

C Überprüfung der funktionalen Anforderungen (Auslegung B)

Wie bereits in Abschnitt 6.1.4 erläutert, existiert neben der in Abschnitt 6.2.2 überprüften Auslegungsvariante des Gesamtfahrstils eine weitere Variante. Diese benutzt nicht nur die Vorhersage der stärksten Clusterzuordnung, sondern berücksichtigt ebenfalls die Zuordnungswerte des betreffenden Datenpunkts zu den übrigen Clustern. Die Zuordnung wird hierbei aus den Abständen des Datenpunkts zu den einzelnen Clusterschwerpunkten berechnet und normiert. Durch die Nutzung aller Zuordnungswerte findet eine genauere Einordnung des Datenpunkts statt.

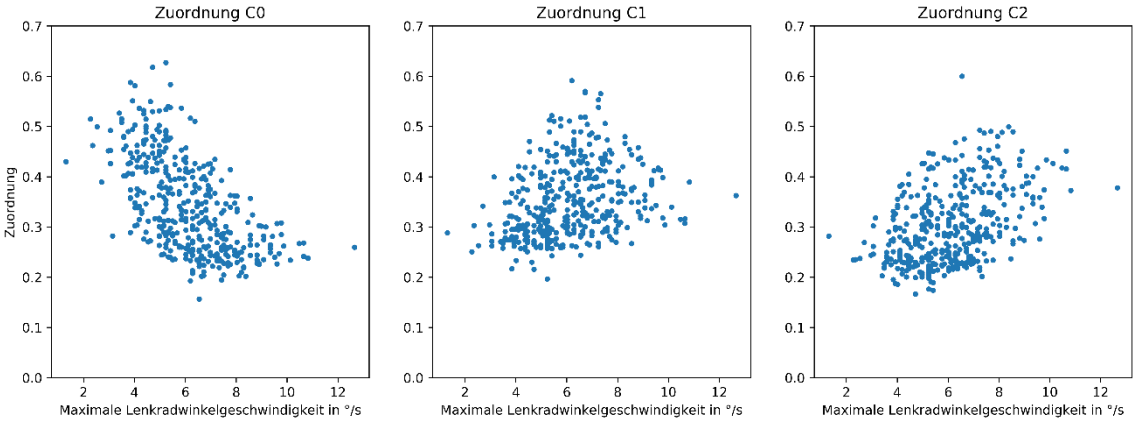
Zu jedem Eingangsdatenpunkt werden daher drei statt einer Ausgabe generiert. Hierdurch ist die Überprüfung der funktionalen Anforderungen, wie in Abschnitt 6.2.2 durchgeführt, nicht möglich. Daher werden die Vorhersagehöhen der Datenpunkte für jedes Cluster einzeln untersucht, wodurch drei getrennte Abbildungen entstehen. Die Datenpunkte werden für jede Abbildung entsprechend der in der Anforderung geforderten Dimensionen geplottet. Zur Visualisierung der Vorhersagehöhen werden unterschiedliche Graustufen genutzt. Als Beispiel ist in Anhang 8-7 die Überprüfung der funktionalen Anforderung L1 des originalen Modells gezeigt. Je dunkler ein Datenpunkt eingefärbt ist, desto stärker ist dessen Zuordnung zum jeweiligen Cluster. Es wird deutlich, dass auch in dieser Darstellung die von L1 geforderte Ordinalität der Cluster zwischen C2 und C1/ C0 vorhanden ist und der Anforderung L1 zumindest nicht widersprochen wird.



Anhang 8-7: Überprüfung der Anforderung L1 mit allen Clusterzuordnungen

Analog zu diesem Beispiel lässt sich L2 überprüfen. Durch das Vorliegen mehrerer kontinuierlicher Ausgangsgrößen ist es nicht möglich die Überprüfung von L3 und L4 mit einem CDF-Diagramm vorzunehmen. Als Alternative wird das jeweilige Merkmal über der Clusterzuordnung für je ein Cluster dargestellt. Die sich ergebende Darstellung zur Überprüfung der Anforderung L3 ist in Anhang 8-8 gegeben. In der Betrachtung des Clusters C0 (vorsichtig) wird der untere Wertebereich der maximalen Lenkradwinkelgeschwindig-

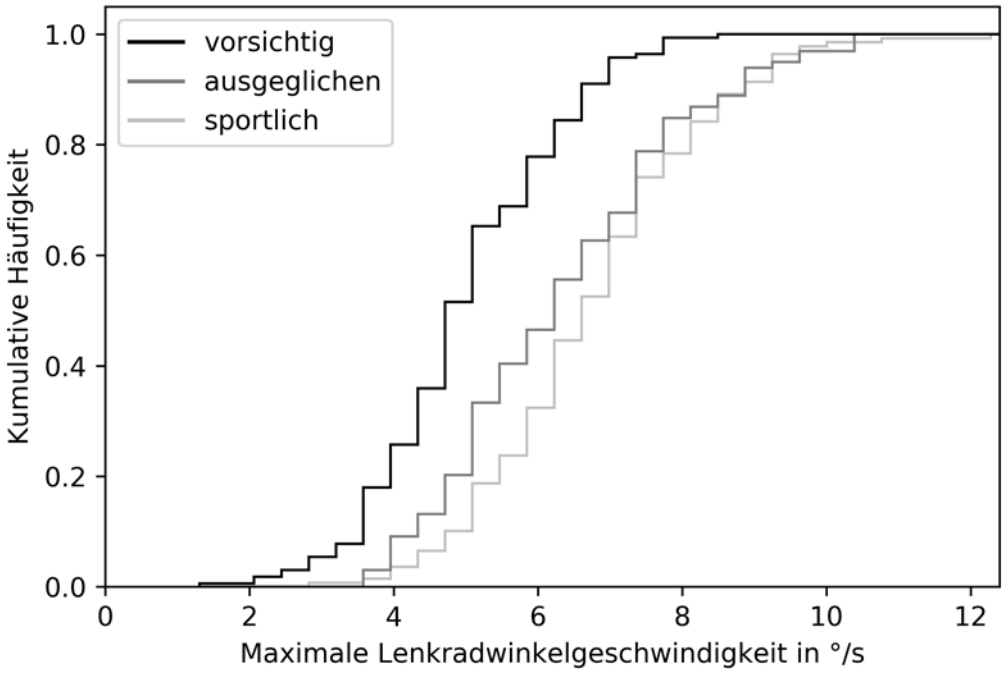
keit stärker dem Cluster C0 zugeordnet, als der mittlere oder höhere Wertebereich. Die Zuordnung C1 (ausgeglichen) zeigt dieses Verhalten für den mittleren Wertebereich und die Zuordnung C2 (sportlich) für den oberen Wertebereich der Lenkradwinkelgeschwindigkeit. Hierdurch ist auch in der Betrachtung aller Ausgänge des k-Means die Anforderung L3 erfüllt.



Anhang 8-8: Überprüfung der Anforderung L3 mit allen Clusterzuordnungen

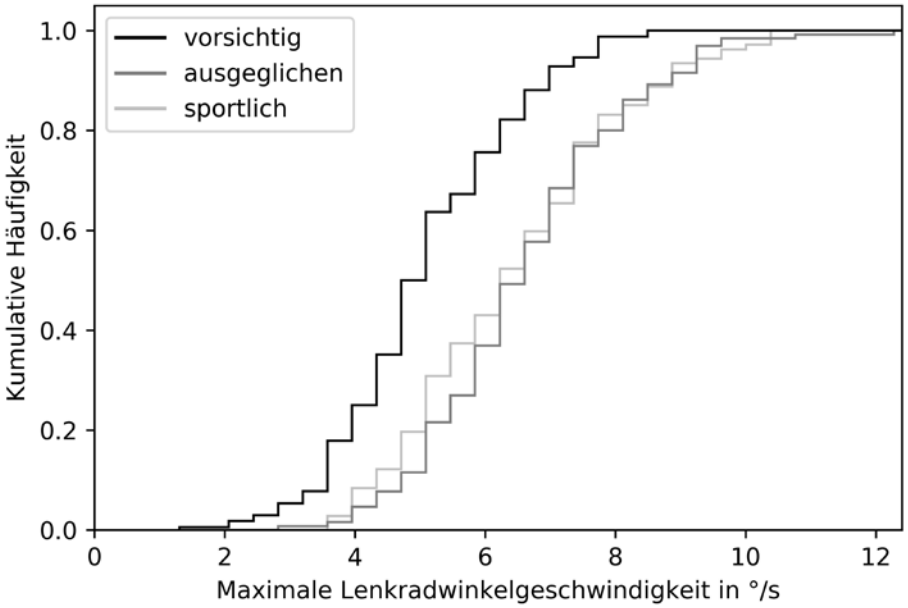
D Überprüfung der Robustheitsanforderungen

D.1 Anforderung DQ1

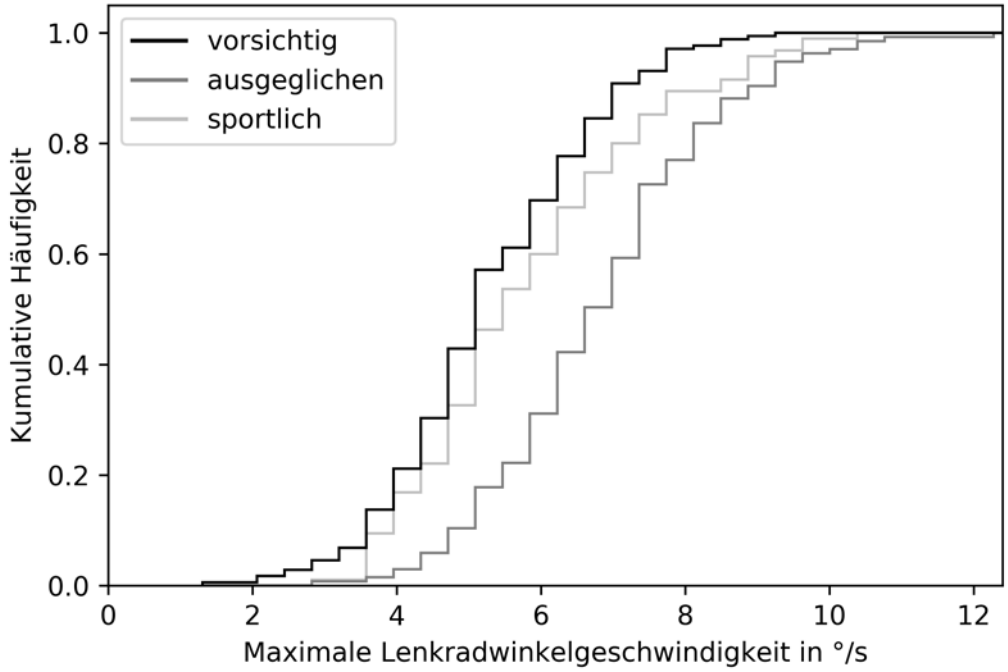


Anhang 8-9: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 4

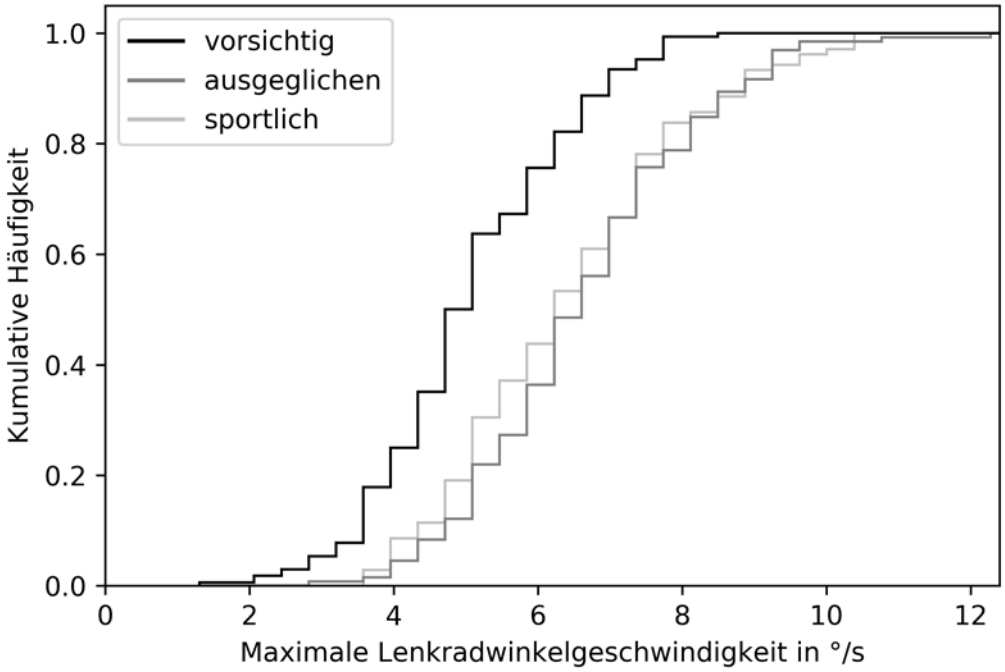
D.2 Anforderung DQ2



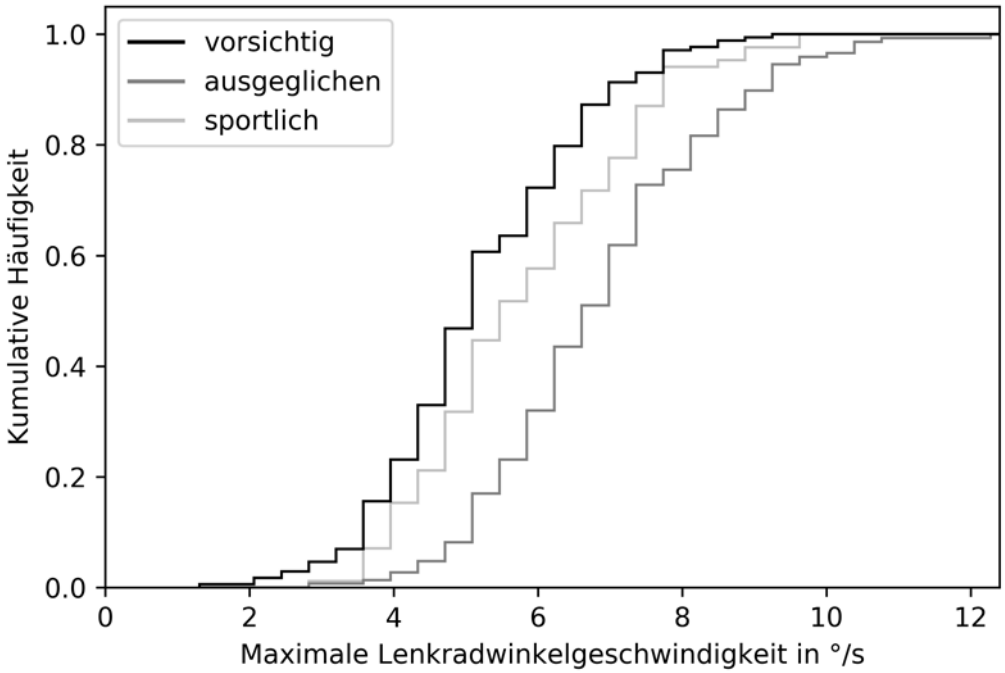
Anhang 8-10: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 1



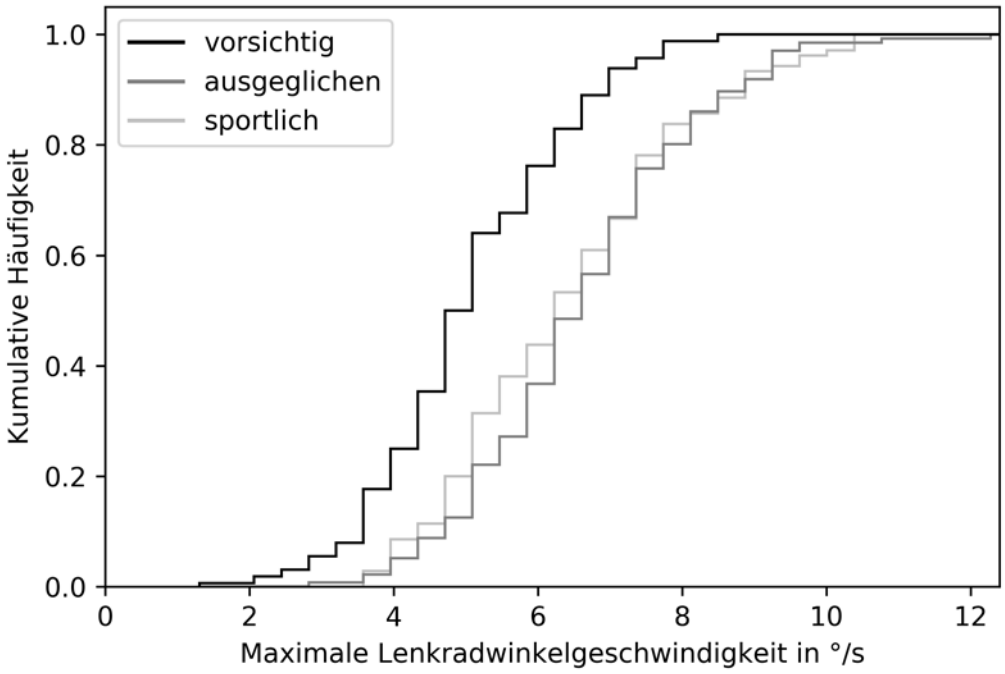
Anhang 8-11: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 2



Anhang 8-12: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 3

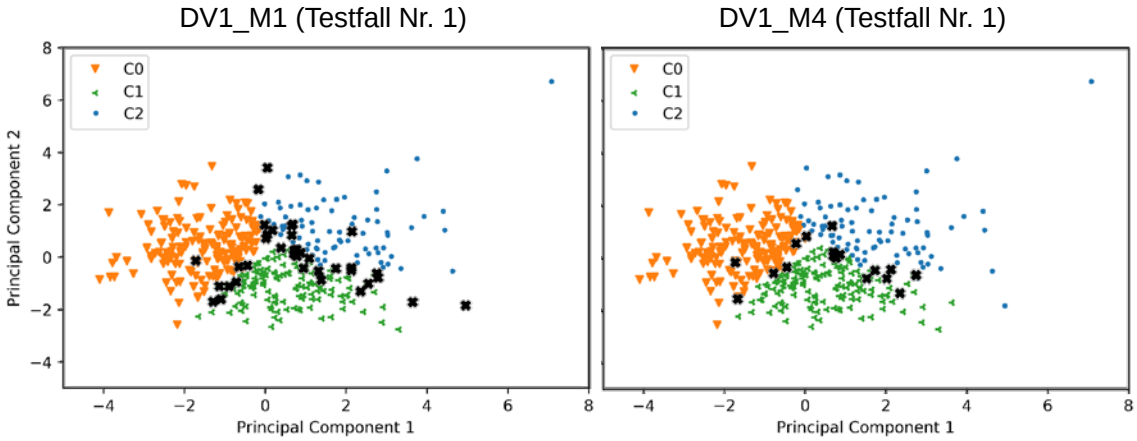


Anhang 8-13: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 4

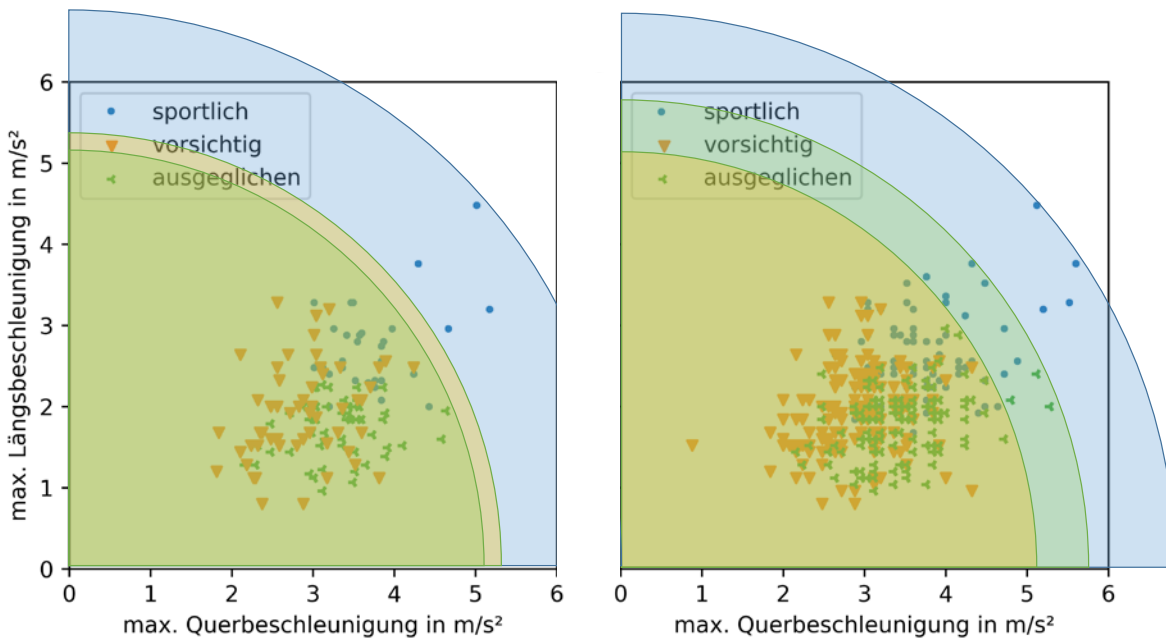


Anhang 8-14: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 5

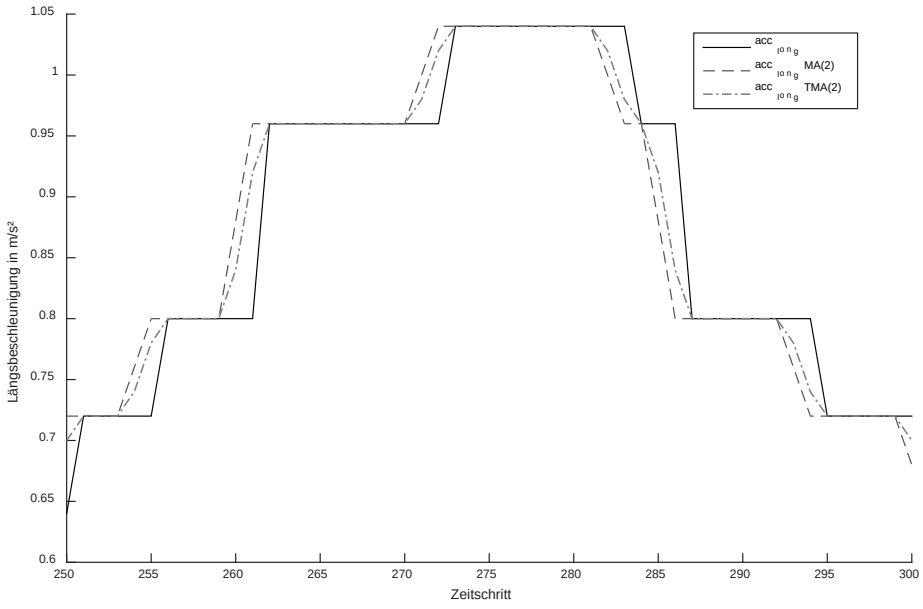
D.3 Anforderung DV1



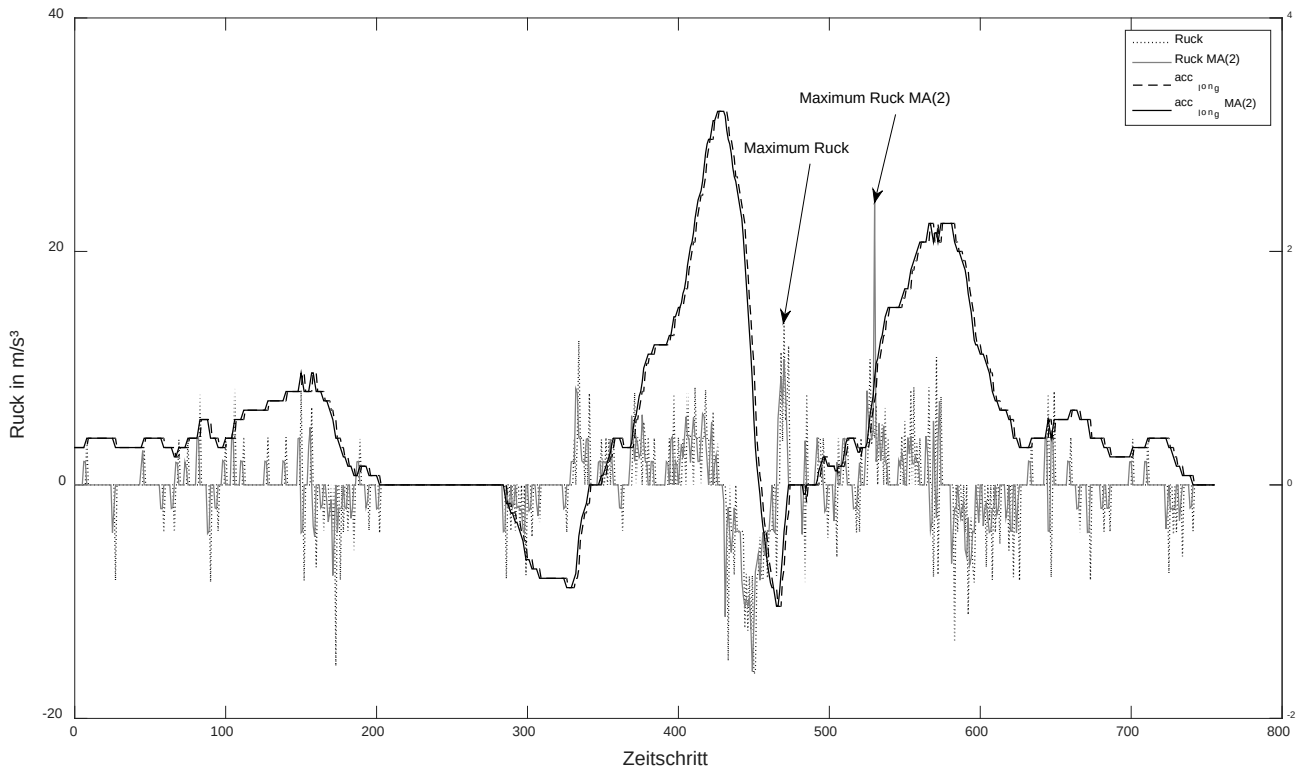
Anhang 8-15: PCA des originalen Datensatzes mit Einfärbung der entsprechenden originalen Clusterzuordnung mit Markierung der nicht-übereinstimmenden Datenpunkte



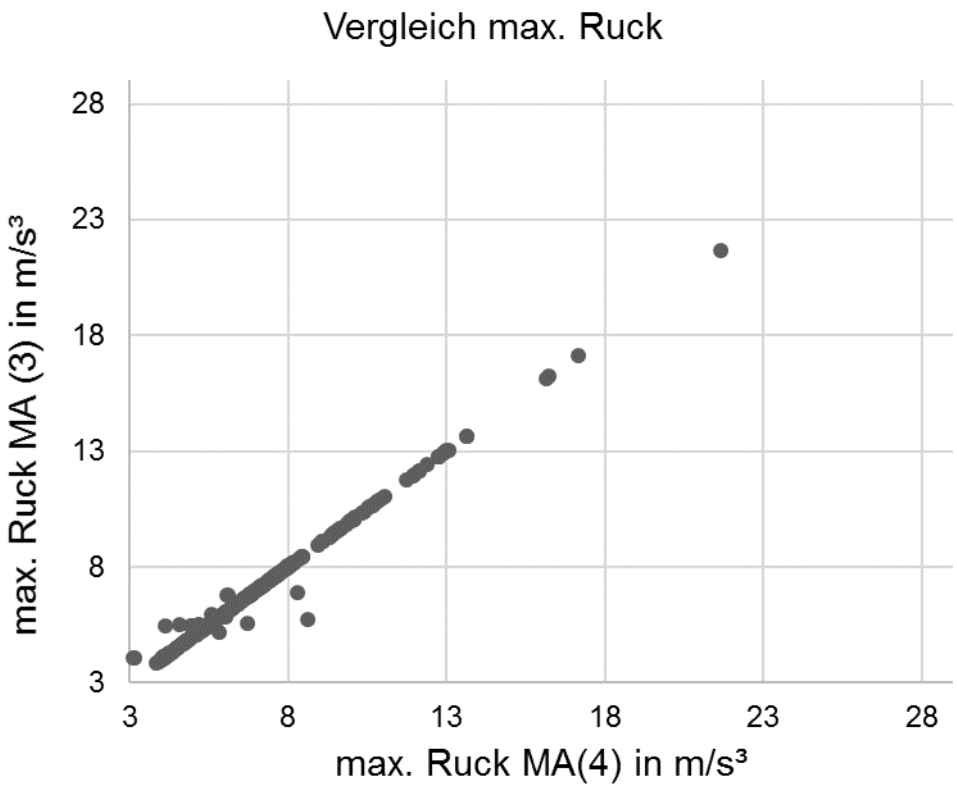
Anhang 8-16: Beispiel der Anforderungserfüllung L2 der Methode DV1_M4 (links: Testfall Nr. 2, rechts: originales Modell mit originalen Daten)



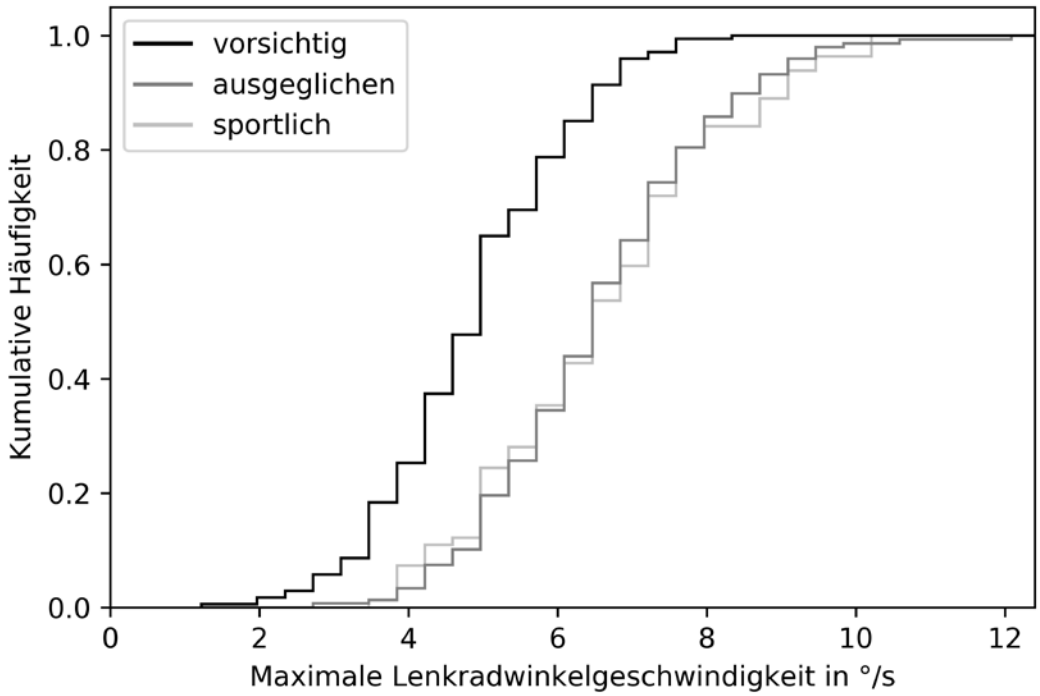
Anhang 8-17: Glättung der Längsbeschleunigung mit MA (2) und TMA (2)



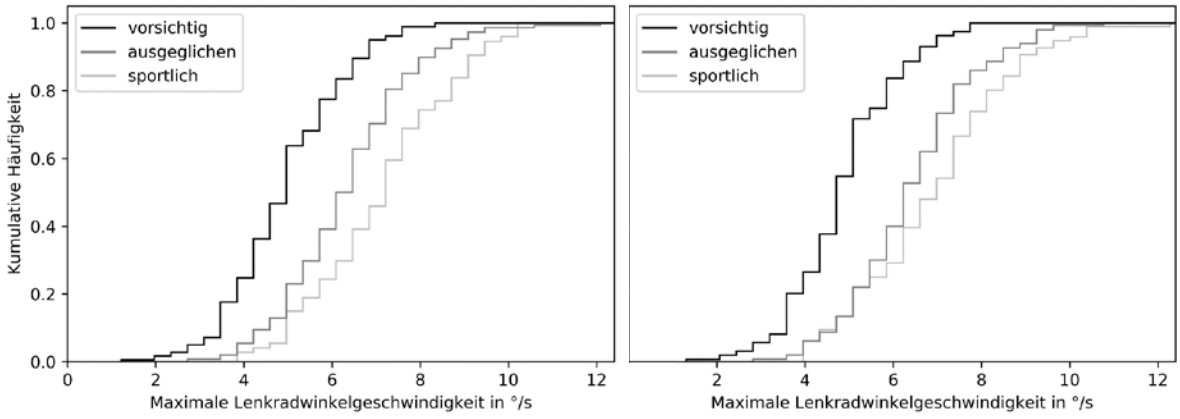
Anhang 8-18: Vergleich Längsbeschleunigung und Ruck original gegenüber geglättet (MA(2)) eines Linksabbiege-Manövers



Anhang 8-19: Zusammenhang zwischen den maximalen Ruckwerten der Testfälle Nr. 2 und Nr. 3

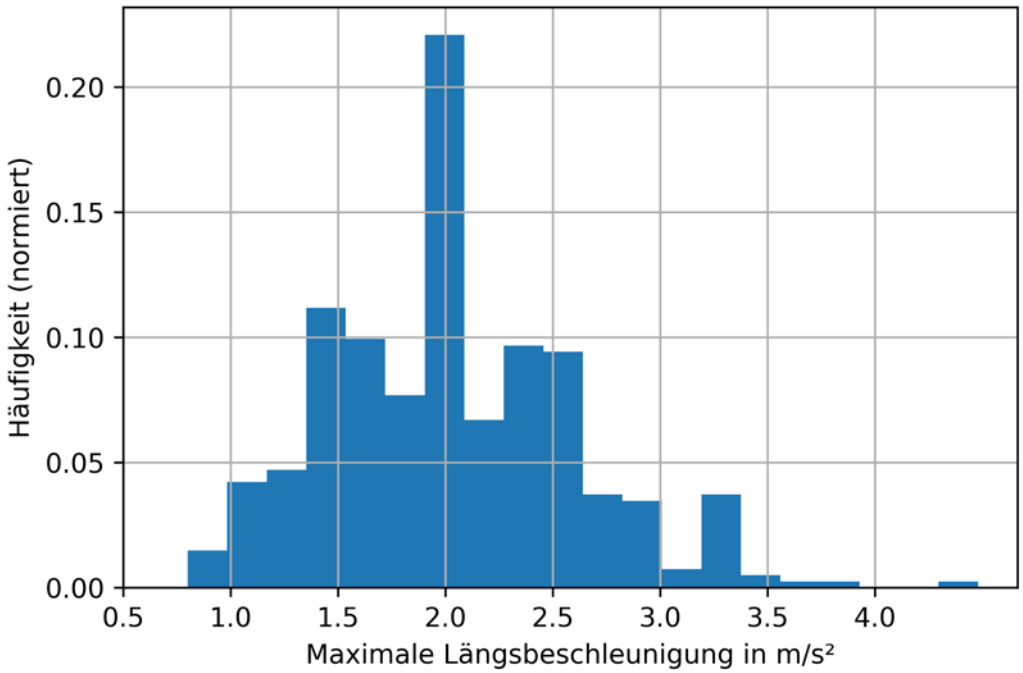


Anhang 8-20: CDF-Darstellung der Lenkradwinkelgeschwindigkeit des Testfalls Nr. 8 (DV1_M5)

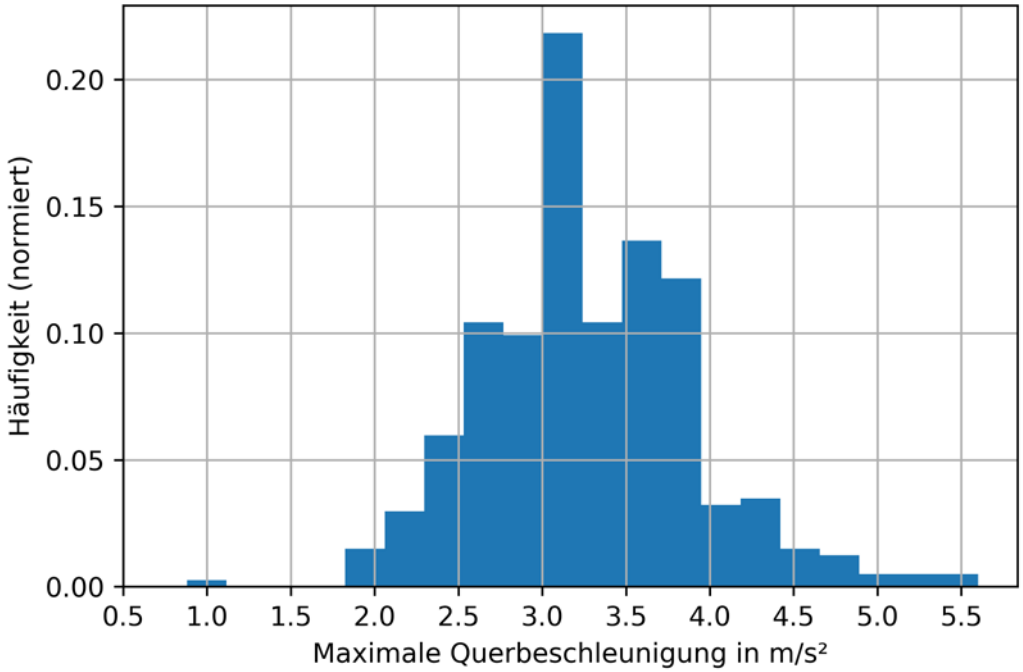


Anhang 8-21: Vergleich der Anforderungserfüllung L3 des Testfalls Nr. 1 (DV1_M5) (links) und des originalen Modells (rechts)

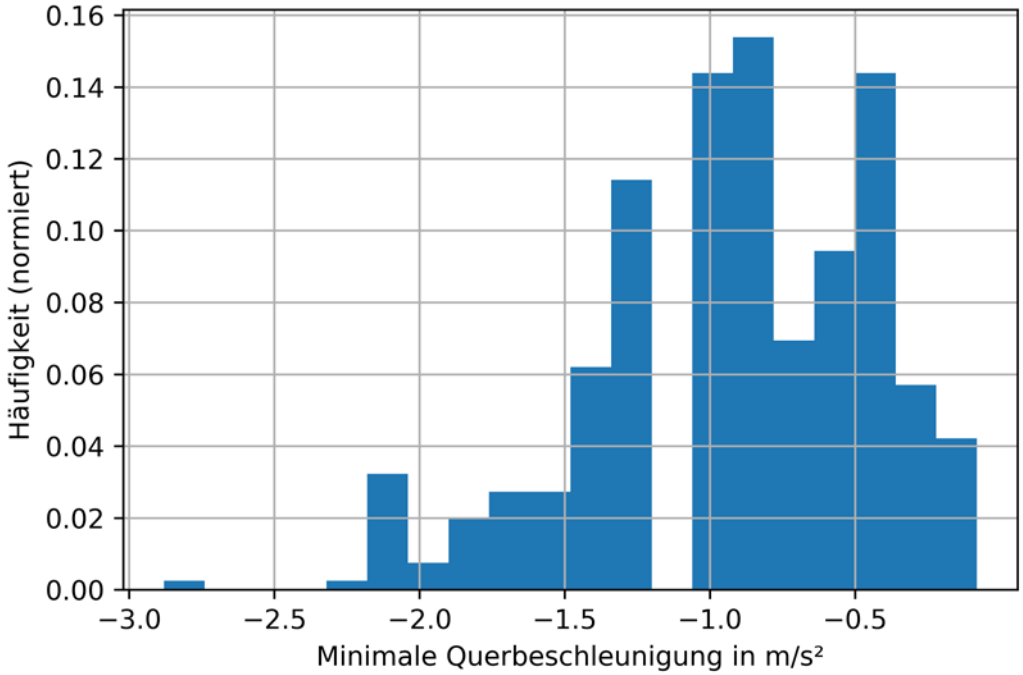
D.4 Anforderung A1



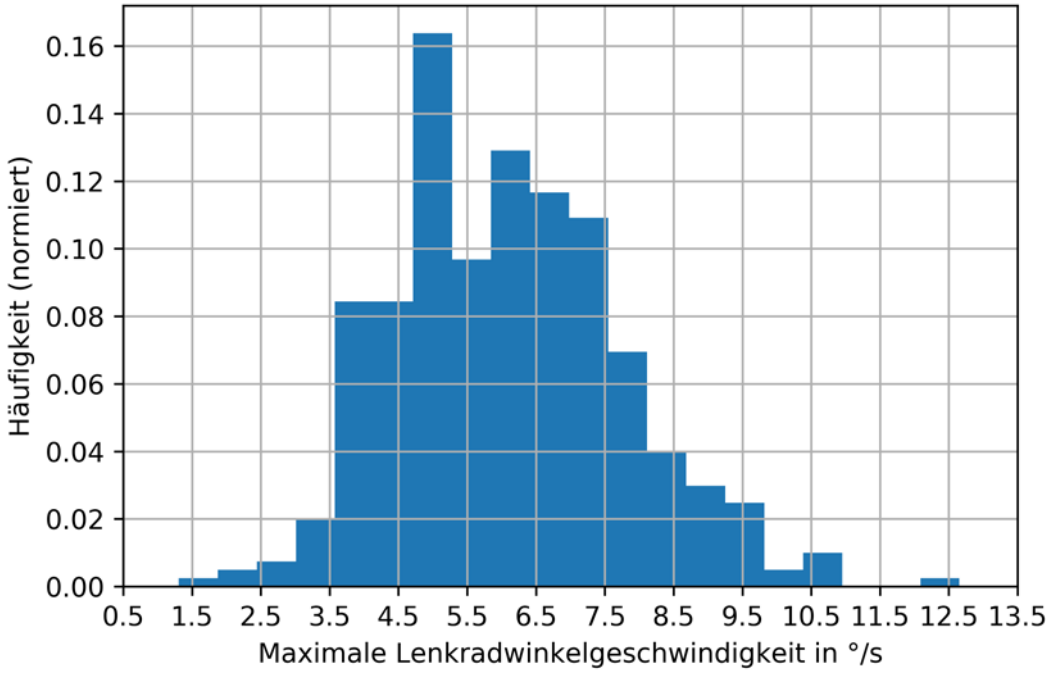
Anhang 8-22: Histogramm des Merkmals maximale Längsbeschleunigung (Bereiche: 20)



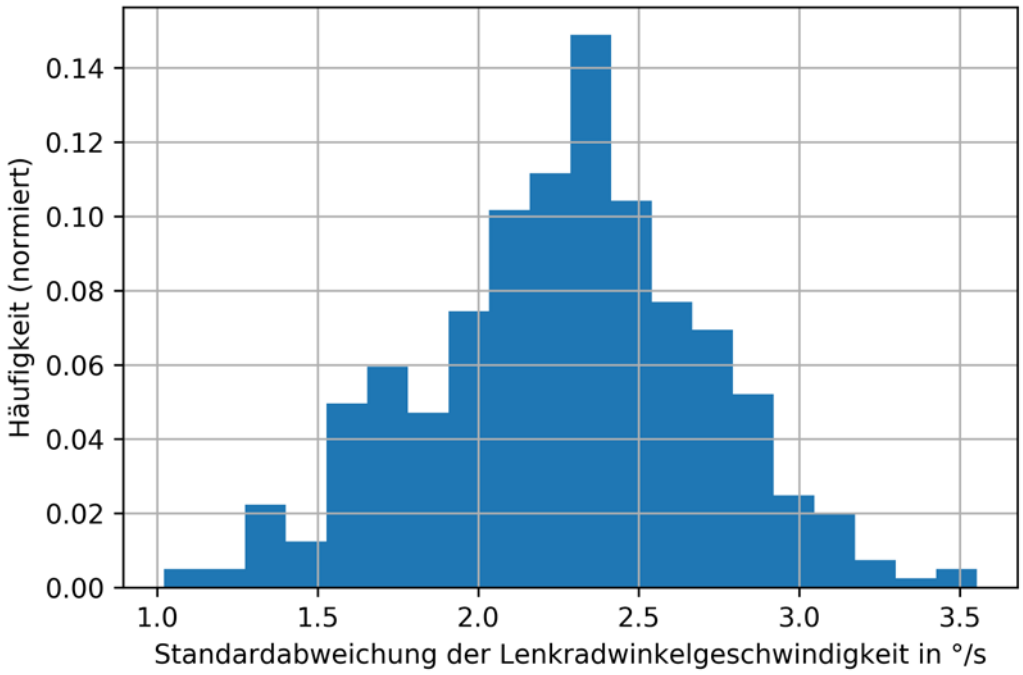
Anhang 8-23: Histogramm des Merkmals maximale Querbeschleunigung (Bereiche: 20)



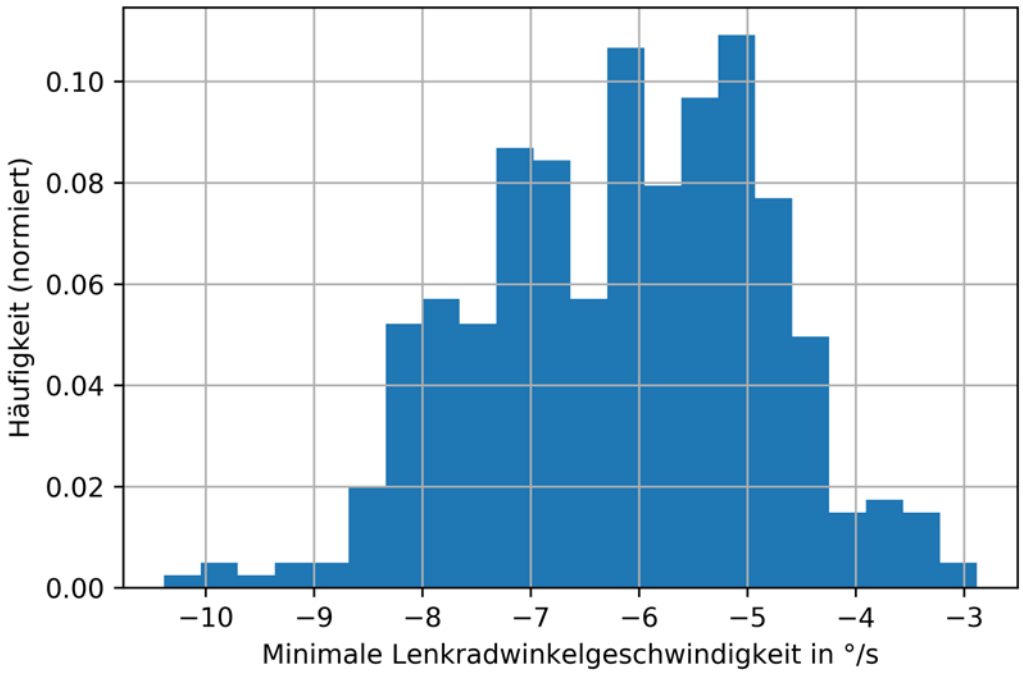
Anhang 8-24: Histogramm des Merkmals minimale Querbeschleunigung (Bereiche: 20)



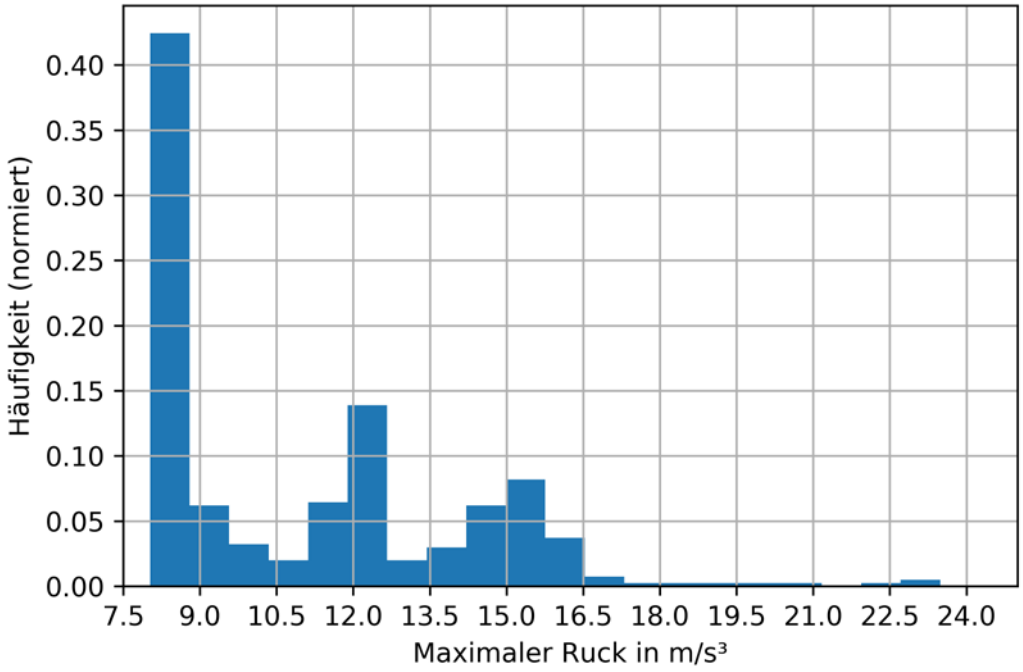
Anhang 8-25: Histogramm des Merkmals maximale Lenkradwinkelgeschwindigkeit (Bereiche: 20)



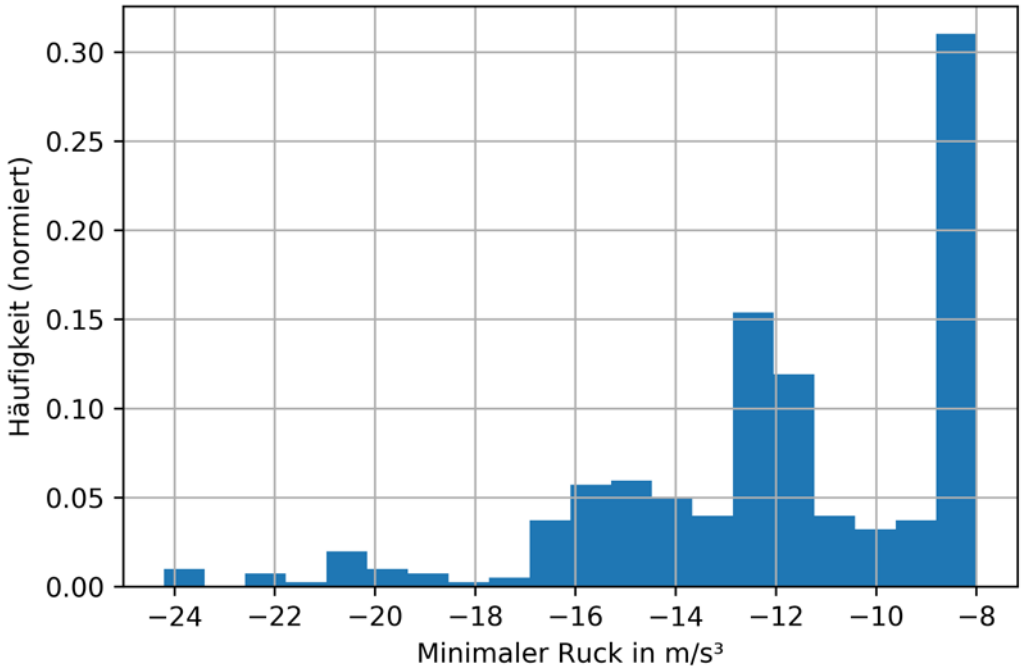
Anhang 8-26: Histogramm des Merkmals Standardabweichung der Lenkradwinkelgeschwindigkeit (Bereiche: 20)



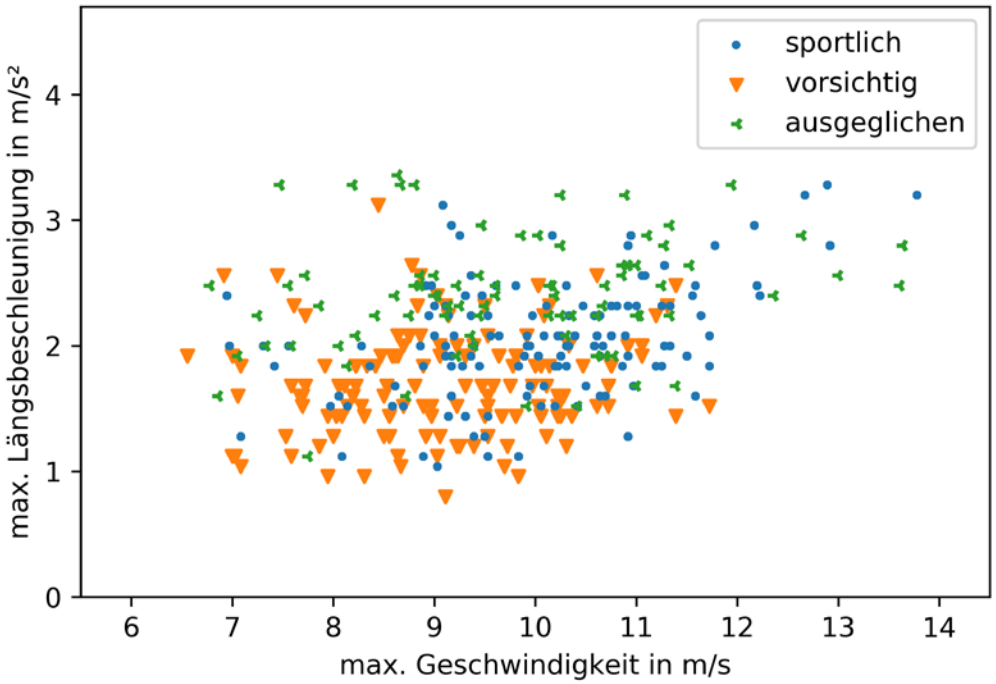
Anhang 8-27: Histogramm des Merkmals Minimale Lenkradwinkelgeschwindigkeit (Bereiche: 22)



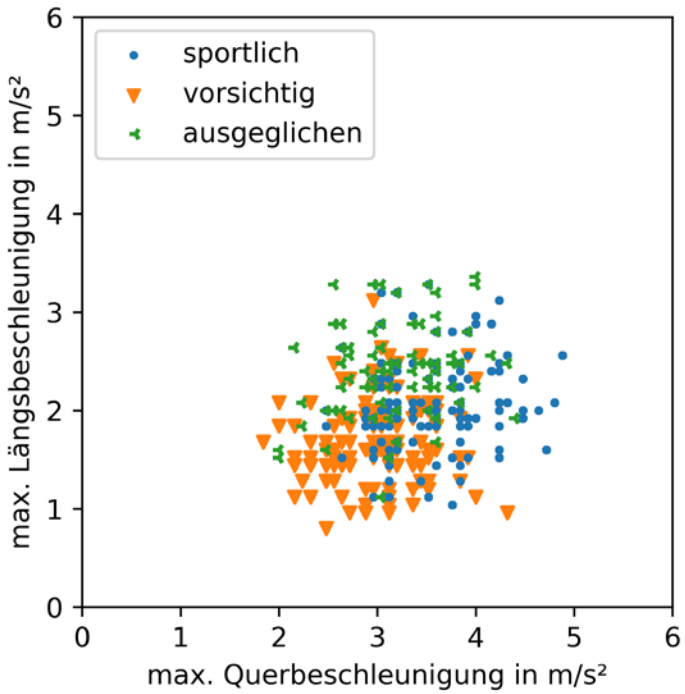
Anhang 8-28: Histogramm des Merkmals Maximaler Ruck (Bereiche: 20)



Anhang 8-29: Histogramm des Merkmals Minimaler Ruck (Bereiche: 20)

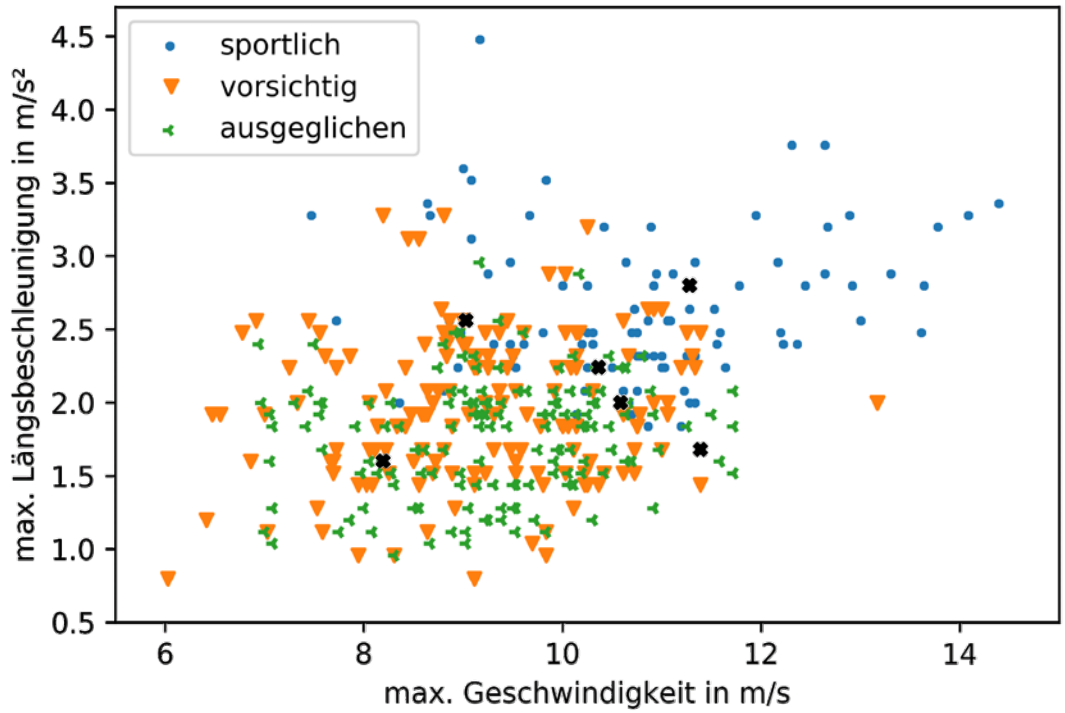


Anhang 8-30: Darstellung der Anforderung L1 des Testfalls Nr. 1

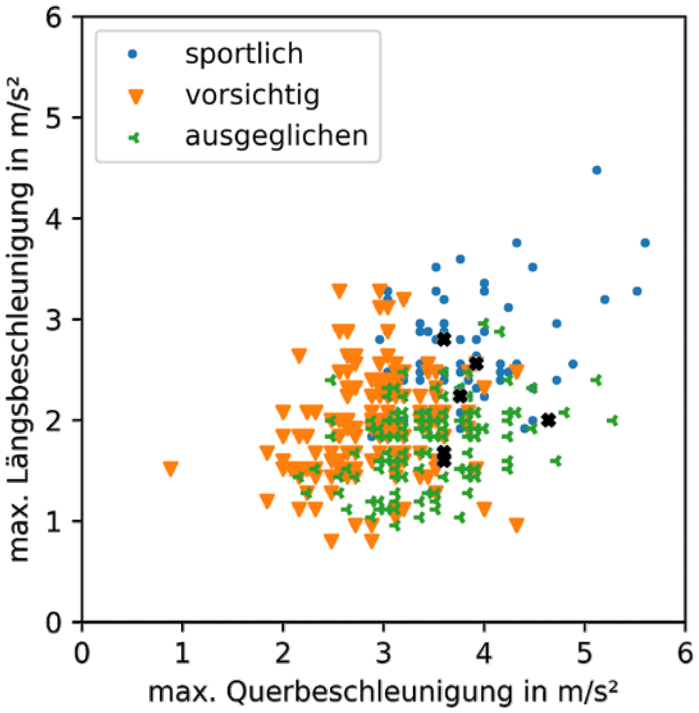


Anhang 8-31: Darstellung der Anforderung L2 des Testfalls Nr. 1

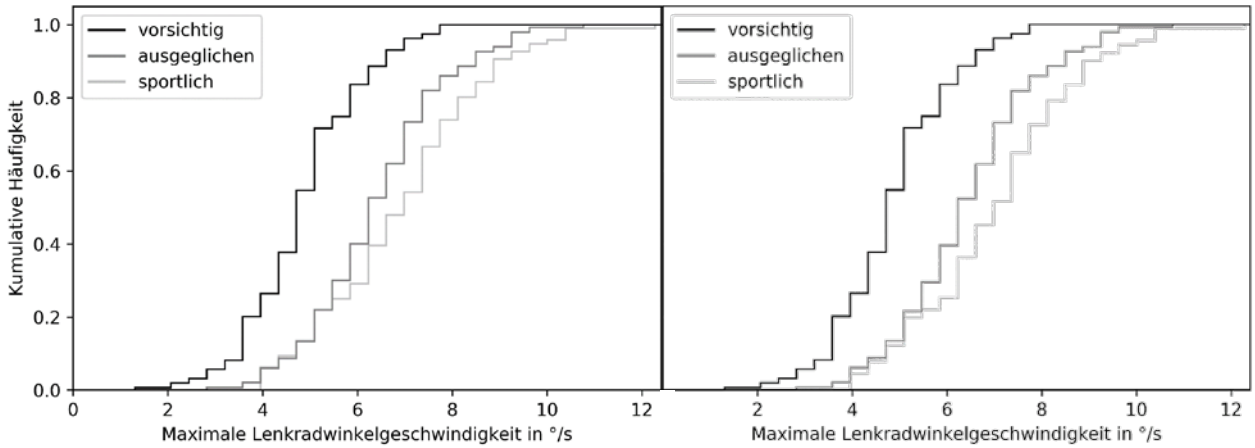
D.5 Anforderung T1



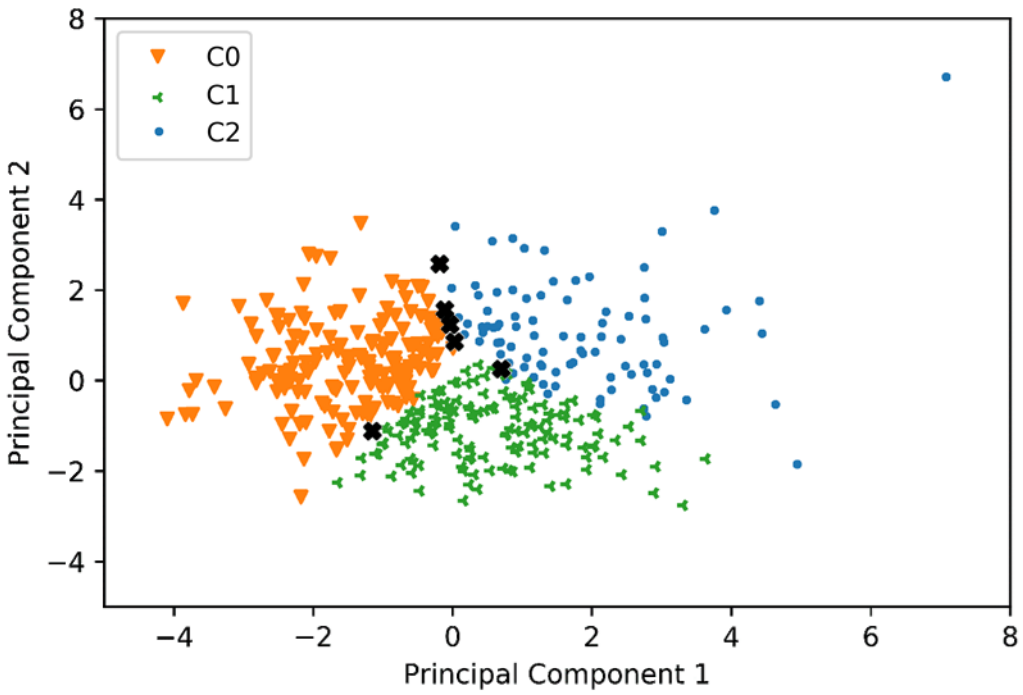
Anhang 8-32: Auffällige Datenpunkte in der Darstellung der Anforderung L1



Anhang 8-33: Auffällige Datenpunkte in der Darstellung der Anforderung L2



Anhang 8-34: Auffällige Datenpunkte in der Darstellung der Anforderung L3 (links mit den Datenpunkten, rechts ohne die Datenpunkte)



Anhang 8-35: Auffällige Datenpunkte in der Darstellung der PCA

D.6 Bestimmung der Signifikanz

Die Beschreibung der Generalisierbarkeit der zur Überprüfung von DQ1 durchgeführten Testfälle Nr. 6 bis 10 beruht auf der Annahme, dass die Wahrscheinlichkeit der Übereinstimmung der Cluster zwischen reduziertem und originalem Modell sehr hoch ist. Die Clusterübereinstimmung wird in den Testfällen durch die Angabe der relativen Übereinstimmung der Clustervorhersage bezogen auf die überprüfte Datenmenge berechnet. Zur

Vereinfachung wird jedoch angenommen, dass die Vorhersagen des reduzierten und des originalen Modells insgesamt übereinstimmen, sobald eine relative Übereinstimmung von 87% vorliegt, da bei dieser Übereinstimmungsrate noch eine Erfüllung der funktionalen Anforderungen vorliegt. Daher lauten die Nullhypothese H_0 sowie die zugehörige Alternativhypothese H_1:

H_0: Die Clustervorhersagen stimmen mit einer Wahrscheinlichkeit von kleiner-gleich 0,95 überein.

H_1: Die Clustervorhersagen stimmen mit einer Wahrscheinlichkeit von größer 0,95 überein.

Damit die Hypothese H_0 statistisch signifikant (oder hochsignifikant) wiederlegbar ist, ist eine bestimmte Mindestanzahl an Testfällen notwendig. Zur Bestimmung der Anzahl an Testfällen ist zunächst zu bestimmen, welcher Verteilung die Ergebnisse des Testfalls folgen. Eine mögliche Verteilung stellt dabei die Binomialverteilung dar. Zur Anwendung der Binomialverteilung sind folgende Annahmen zu erfüllen:

- Jeder Testfall besitzt lediglich die möglichen Ausgänge „Erfolg“ und „Misserfolg“.
- Die Wahrscheinlichkeit für Erfolg bleibt über alle Testfälle gleich.
- Die Testfälle sind voneinander unabhängig.
- Die Testfallauswahl erfolgt zufällig.[350]

Die erste Annahme trifft jedoch eigentlich nicht auf die vorliegenden Testfälle zu, da die Übereinstimmung der Clustervorhersage in kontinuierlichen Werten angegeben wird. Daher gilt die bereits beschriebene Vereinfachung. Unter dieser Voraussetzung ist es möglich die Binomialverteilung anzunehmen. Die Wahrscheinlichkeitsfunktion einer Binomialverteilung wird durch die Anzahl an Stichproben (bzw. im vorliegenden Fall Testfälle) n_T, durch die Anzahl der Erfolge n_S und die Trefferwahrscheinlichkeit p beschrieben.[350]

$$f(n_T, n_S, p) = \binom{n_T}{n_S} p^{n_S} (1-p)^{n_T - n_S} \tag{8.1}$$

Zur Berechnung der minimal notwendigen Anzahl an Testfällen wird angenommen, dass alle Testfälle erfolgreich sind, d.h. $n_T = n_S$ gilt. Die Wahrscheinlichkeit für Erfolg der Testfälle wird, wie in H_0 definiert, auf 0,95 festgelegt. Die Formel (8.1) vereinfacht sich zu:

$$f(n_T) = 0{,}95^{n_T} \tag{8.2}$$

[350] Vgl. Henze, N.: Die Binomialverteilung und die Multinomialverteilung (2018), S. 142 ff.

Um eine statistisch signifikante Widerlegung der Nullhypothese zu erlangen, hat $f(n_T) < 0{,}05$ zu sein, eine hoch signifikante Widerlegung liegt bei $< 0{,}001$ vor. Nach Formel (8.2) wird dies bei $n_T = 59$ bzw. $n_T = 135$ erreicht. Sind bei $n_T = 59$ alle Testfälle positiv, d.h. stimmen alle Clustervorhersagen der Testfälle mind. zu 87% überein, liegt die Wahrscheinlichkeit, dass dieser Fall eintritt bei unter 5% gegeben der Annahme, dass ein Risiko von über 5% vorliegt, dass die Clustervorhersagen durch Zufall hervorgehen. Hierdurch wird H_0 signifikant wiederlegt. Werden weniger als 59 Testfälle durchgeführt, ist es jedoch nicht möglich H_0 signifikant zu widerlegen.

D.7 Anwendbarkeit Auslegung B

Analog zur Überprüfung der prinzipiellen Anwendbarkeit des dritten Schritts beim Vorliegen mehrerer Ausgangsgrößen wird diese Überprüfung ebenfalls im Rahmen der Robustheitsanforderungen durchgeführt. Hierzu wird statt der Auslegungsvariante A des Systems, welche lediglich die Vorhersage des stärksten Clusters pro Datenpunkt in der Erstellung des Gesamtfahrstils berücksichtigt, die Auslegungsvariante B betrachtet, die die Zuordnung eines Datenpunkts zu allen Clustern für die Erstellung des Gesamtfahrstils nutzt.[351] Die Höhe der Zuordnung eines Datenpunkts zu allen Clustern ist normiert und wird aus der Inversen des Abstands des Datenpunkts zum jeweiligen Clusterschwerpunkt berechnet. In Auslegungsvariante A wird aus diesen normierten Zuordnungen der Cluster vorhergesagt bzw. zum Modul der Gesamtfahrstilberechnung weitergeleitet, der dem Datenpunkt am nächsten ist bzw. der die höchste Zuordnung besitzt.

Das Vorgehen der Testfallableitung ändert sich zu Auslegungsvariante A nicht, lediglich die Evaluation des resultierenden Modells ist anzupassen. Die Bestimmung der Übereinstimmung der Vorhersage zwischen dem aus dem Testfall resultierenden Modell und dem originalen Modell wird bei einer diskreten Ausgangsgröße, wie in Auslegungsvariante A, durch den direkten Vergleich des Ausgangswerte vorgenommen. Allerdings besitzt diese Evaluationsmethode den Nachteil, dass die Veränderungen der Vorhersage innerhalb der diskreten Stufen nicht offenbart, wodurch zusätzlich die funktionalen Anforderungen miteinander verglichen werden, um deren Einhaltung sicherzustellen. Zusätzlich tritt das Problem auf, dass Datenpunkte, die nahe an Clustergrenzen liegen, d.h. deren stärkstes Cluster etwas mehr als 0,33 normierter Zuordnung besitzt, schon bei minimaler Änderung der Clusterschwerpunkte in die „falsche“ Richtung eine veränderte Vorhersage erhalten. Daher werden die nicht-übereinstimmenden Datenpunkte in Auslegungsvariante A einer genaueren Analyse unterzogen, um die Ursache der fehlenden Übereinstimmung zu identifizieren.

Durch die Nutzung aller Clusterzuordnungen eines Datenpunkts wird diesem Problem begegnet. Auch die Vorhersageänderung innerhalb der einzelnen Cluster wird hierdurch offensichtlich. Allerdings resultiert jede minimale Änderung der Clusterschwerpunkte in einer betragsmäßig veränderten Zuordnung eines Datenpunkts zu jedem Cluster, wodurch der Vergleich der Vorhersagen zweier Modelle erschwert wird. Es ergeben sich prinzipiell drei Möglichkeiten der Evaluation:

- 1. Möglichkeit:

  Gesamtschwellwert: Berechnung der Differenzen der Zuordnung jedes Clusters pro Datenpunkt mit anschließender Addition aller betragsmäßigen Differenzen pro

[351] Siehe Abschnitt 6.1.4.

Datenpunkt. Durch einen Gesamtschwellwert wird festgelegt, ob diese Gesamtdifferenz als Abweichung/ Fehler gewertet wird oder nicht.

- 2. Möglichkeit:

  Einzelschwellwert: Berechnung der Differenzen der Zuordnung jedes Clusters pro Datenpunkt. Durch einem Schwellwert für jedes Cluster wird festgelegt, ob die Differenz als Abweichung/ Fehler gewertet wird oder nicht.

- 3. Möglichkeit:

  Kombination beider Methoden: Berechnung der Differenzen der Zuordnung jedes Clusters pro Datenpunkt. Die Nutzung eines Einzelschwellwerts detektiert relativ große Differenzen innerhalb einer Clusterzuordnung. Überschreiten zwei Cluster diesen Schwellwert, wird dieser Datenpunkt als „Abweichung“ zwischen originalem und resultierenden Modell gezählt. Die Nutzung eines Gesamtschwellwerts detektiert, ob sich die generelle Lage eines Datenpunkts hinsichtlich der Clusterschwerpunkte stark verändert.

Die Ergebnisse aller aufgezählten Möglichkeiten hängen natürlich von den genutzten Schwellwerten ab. Dieser ist je nach der Sicherheitsrelevanz der Auswirkung der Veränderung der Zuordnungshöhe zu bestimmen. Der zusätzliche Vergleich der funktionalen Anforderungen für jeden Testfall bei Vorliegen mehrerer Ausgangsgrößen wird in Anhang C gezeigt. Prinzipiell ist daher die Anwendbarkeit des vierten Schrittes auch bei Vorliegen mehrerer Ausgangsgrößen gegeben.

Literaturverzeichnis

Akarachai, A.; Daricha, S.: Avoiding Local Minima (2007)
Akarachai, Atakulreka; Daricha, Sutivong: Avoiding Local Minima in Feedforward Neural Networks by Simultaneous Learning, in: Orgun, Mehmet A.; Thornton, John (Hrsg.): AI 2007, Advances in Artificial Intelligence, Springer, Berlin, Heidelberg, 2007

Alpaydin, E.: Introduction to machine learning (2004)
Alpaydin, Ethem: Introduction to machine learning, MIT Press, Cambridge, Mass., 2004

Arroyo, R. et al.: Fusion and binarization of CNN features (2016)
Arroyo, Roberto; Alcantarilla, Pablo F.; Bergasa, Luis M.; Romera, Eduardo: Fusion and binarization of CNN features for robust topological localization across seasons, in: IROS 2016, IEEE, 2016

Arthur, D.; Vassilvitskii, S.: k-means++: the advantages of careful seeding (2007)
Arthur, David; Vassilvitskii, Sergej: k-means++: the advantages of careful seeding, in: Society for Industrial and Applied Mathematics Philadelphia (Hrsg.): Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007

Awad, M.; Khanna, R.: Efficient Learning Machines (2015)
Awad, Mariette; Khanna, Rahul: Efficient Learning Machines, Imprint: Apress, Berkeley, CA, 2015

Backhaus, K. et al.: Neuronale Netze (2018)
Backhaus, Klaus; Erichson, Bernd; Plinke, Wulff; Weiber, Rolf: Neuronale Netze, in: Backhaus, Klaus et al. (Hrsg.): Multivariate Analysemethoden, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018

Balzert, H.: Lehrbuch der Softwaretechnik (1998)
Balzert, Helmut: Lehrbuch der Softwaretechnik, 2. Auflage, 1998

Balzert, H.: Nichtfunktionale Anforderungen (2011)
Balzert, Helmut: Nichtfunktionale Anforderungen, in: Balzert, Helmut *1. (Hrsg.): Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, Spektrum, Heidelberg [u.a.], 2011

Batch learning (2017) Batch learning, in: Sammut, Claude; Webb, Geoffrey I. (Hrsg.): Encyclopedia of Machine Learning and Data Mining, Springer US, Boston, MA, 2017

Batista, J. P.: A Real-Time Driver Visual Attention Monitoring System (2005)
Batista, Jorge P.: A Real-Time Driver Visual Attention Monitoring System, in: Marques, Jorge S.; La Pérez de Blanca, Nicolás; Pina, Pedro (Hrsg.): Pattern Recognition and Image

Analysis, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005

Behera, R.; Das, K.: A Survey on Machine Learning (2017)
Behera, Rabi; Das, Kajaree: A Survey on Machine Learning, in: International Journal of Innovative Research in Computer and Communication Engineering, Jahrgang 2, 2017

Berg, G. et al.: Vehicle in the Loop (2016)
Berg, Guy; Nitsch, Verena; Färber, Berthold: Vehicle in the Loop, in: Winner, Hermann et al. (Hrsg.): Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort, Springer International Publishing, Cham, 2016

Bergadano, F.: The Problem of Induction and Machine Learning (1991)
Bergadano, Ferdinand: The Problem of Induction and Machine Learning, in: Kaufman, Morgan (Hrsg.): IJCAI '91, San Francisco, 1991

Bishop, C. M.: Pattern recognition and machine learning (2006)
Bishop, Christopher M.: Pattern recognition and machine learning, Information science and statistics, Springer, New York, NY, 2006

Bojarski, M. et al.: End to end learning for self-driving cars (2016)
Bojarski, Mariusz; Del Testa, Davide; Dworakowski, Daniel; Firner, Bernhard; Flepp, Beat; Goyal, Prasoon; Jackel, Lawrence D.; Monfort, Mathew; Muller, Urs; Zhang, Jiakai; others: End to end learning for self-driving cars, in: arXiv preprint arXiv:1604.07316, 2016

Bossdorf-Zimmer, J. et al.: Fingerprint des Fahrers (2011)
Bossdorf-Zimmer, Janine; Kollmer, Hermann; Henze, Roman; Küçükay, Ferit: Fingerprint des Fahrers zur Adaption von Assistenzsystemen, in: ATZ-Automobiltechnische Zeitschrift (3), Jahrgang 113, S. 226–231, 2011

Bouzouraa, M. E.: Verfahren zum Betreiben einer Mensch-Maschine-Schnittstelle (2014)
Bouzouraa, Mohamed E.: Verfahren zum Betreiben einer Mensch-Maschine-Schnittstelle eines Kraftfahrzeugs und zugehöriges Kraftfahrzeug, Audi AG, Patent DE102014019105 A1, 2014

Brain4Cars (2016) Brain4Cars; http://brain4cars.com/, 2016, Zugriff 18.09.2017

Brockmann, M.: Code of Practice for the Design and Evaluation of ADAS (2009)
Brockmann, Martin: Code of Practice for the Design and Evaluation of ADAS, 2009

Bronshtein, A.: Train/Test Split and Cross Validation in Python (2017)
Bronshtein, Adi: Train/Test Split and Cross Validation in Python; https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6, 2017, Zugriff 03.03.2019

Brownlee, J.: How to Prepare Data For Machine Learning (2013)
Brownlee, Jason: How to Prepare Data For Machine Learning;

https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/, 2013, Zugriff 14.01.2019

Brownlee, J.: How Much Training Data is Required for Machine Learning? (2017)
Brownlee, Jason: How Much Training Data is Required for Machine Learning?; https://machinelearningmastery.com/much-training-data-required-machine-learning/, 2017, Zugriff 04.02.2018

Brownlee, J.: What is the Difference Between a Batch and an Epoch in a Neural Network? (2018)
Brownlee, Jason: What is the Difference Between a Batch and an Epoch in a Neural Network?; https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/, 2018, Zugriff 14.04.2019

Bubb, H.: Wie viele Probanden braucht man für allgemeine Erkenntnisse aus Fahrversuchen? (2003)
Bubb, Heiner: Wie viele Probanden braucht man für allgemeine Erkenntnisse aus Fahrversuchen?, in: Landau, Kurt; Winner, Hermann (Hrsg.): Fahrversuche mit Probanden - Nutzwert und Risiko, Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik Nr. 557, VDI-Verl., Düsseldorf, 2003

Budhiraja, A.: Learning Less to Learn Better—Dropout in (Deep) Machine learning (2016)
Budhiraja, Amar: Learning Less to Learn Better—Dropout in (Deep) Machine learning; https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5, 2016, Zugriff 08.03.2019

Bundesministerium der Justiz und für Verbraucherschutz: ProdHaftG (1989)
Bundesministerium der Justiz und für Verbraucherschutz Gesetz über die Haftung für fehlerhafte Produkte, 1989

Bundesministerium der Justiz und für Verbraucherschutz: ProdSG (2011)
Bundesministerium der Justiz und für Verbraucherschutz Gesetz über die Bereitstellung von Produkten auf dem Markt, 2011

Bundesministerium der Justiz und für Verbraucherschutz: FZV (2011)
Bundesministerium der Justiz und für Verbraucherschutz Verordnung über die Zulassung von Fahrzeugen zum Straßenverkehr (Fahrzeug-Zulassungsverordnung – FZV), 2011

Burges, C. J.; Crisp, D. J.: Uniqueness of the SVM solution (2000)
Burges, Christopher J. C.; Crisp, David J.: Uniqueness of the SVM solution, in: Advances in neural information processing systems, 2000

Burke, J.: Vorlesungsunterlagen, Linear Optimization (2018)
Burke, Jim: Linear Optimization, Vorlesungsunterlagen
University of Washington, Washington, 2018

Burton, S. et al.: Case for Safety of Machine Learning (2017)
Burton, Simon; Gauerhof, Lydia; Heinzemann, Christian: Making the Case for Safety of Machine Learning in Highly Automated Driving, in: Tonetta, Stefano; Schoitsch, Erwin; Bitsch, Friedemann (Hrsg.): Computer safety, reliability, and security, LNCS sublibrary. SL 2, Programming and software engineering Nr. 10489, Springer, Cham, Switzerland, 2017

Burton, S.; Bürkle, L.: Making the Case for Safety of Machine Learning (2017)
Burton, Simon; Bürkle, Lutz: Making the Case for Safety of Machine Learning applied to Automated Driving, in: Haus der Technik e.V. (Hrsg.): Aktive Sicherheit und automatisiertes Fahren, Essen, 2017

Butakov, V.; Ioannou, P.: Personalized Driver/Vehicle Lane Change (2015)
Butakov, Vadim; Ioannou, Petros: Personalized Driver/Vehicle Lane Change Models for ADAS, in: IEEE Transactions on Vehicular Technology (10), Jahrgang 64, S. 4422–4431, 2015

Carmona, J. et al.: Analysis of Aggressive Driver Behaviour using Data Fusion (2016)
Carmona, Juan; García, Fernando; Miguel, Miguel Á. de; La Escalera, Arturo de; Armingol, José M.: Analysis of Aggressive Driver Behaviour using Data Fusion, in: Helfert, Markus; Gusikhin, Oleg (Hrsg.): VEHITS 2016, SCITEPRESS - Science and Technology Publications Lda, Setúbal, Portugal, 2016

Chakarov, A. et al.: Debugging Machine Learning Tasks (2016)
Chakarov, Aleksandar; Nori, Aditya; Rajamani, Sriram; Sen, Shayak; Vijaykeerthy, Deepak: Debugging Machine Learning Tasks, 2016

Chawla, N. V.: Data Mining for Imbalanced Datasets (2010)
Chawla, Nitesh V.: Data Mining for Imbalanced Datasets, in: Maimon, Oded Z. (Hrsg.): Data mining and knowledge discovery handbook, Springer, New York [u.a.], 2010

Chen, C. et al.: DeepDriving (2015)
Chen, Chenyi; Seff, Ari; Kornhauser, Alain; Xiao, Jianxiong: DeepDriving, in: Proceedings of the IEEE International Conference on Computer Vision, 2015

Chen, X. et al.: A learning model for personalized adaptive cruise control (2017)
Chen, Xin; Zhai, Yong; Lu, Chao; Gong, Jianwei; Wang, Gang: A learning model for personalized adaptive cruise control, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017

Copeland, M.: Difference Between AI, Machine Learning, and Deep Learning (2016)
Copeland, Michael: What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?; https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/, 2016, Zugriff 12.12.2016

DeAmbroggi, L.: Artificial intelligence in automotive (2016)
DeAmbroggi, Luca: Artificial intelligence in automotive, 2016

DeAmbroggi, L.: Artificial Intelligence Systems (2016)
DeAmbroggi, Luca: Artificial Intelligence Systems for Autonomous Driving on the Rise, IHS Says, 2016, Zugriff 30.11.2016

Desjardins, C.; Chaib-draa, B.: Cooperative Adaptive Cruise Control (2011)
Desjardins, C.; Chaib-draa, B.: Cooperative Adaptive Cruise Control, in: IEEE Transactions on Intelligent Transportation Systems (4), Jahrgang 12, S. 1248–1260, 2011

Di, W. et al.: Deep Learning Essentials (2018)
Di, Wei; Bhardwaj, Anurag; Wei, Jianing: Deep Learning Essentials, Packt Publishing, Birmingham, 2018

Dokania, P. et al.: Online lane change intention prediction (2013)
Dokania, Puneet; Perrollaz, Mathias; Lefevre, Stephanie; Laugier, Christian: Learning-based approach for online lane change intention prediction, in: IEEE Intelligent Vehicles Symposium 2013, 2013

Dong, G.; Liu, H.: Preliminaries and Overview (2018)
Dong, Guozhu; Liu, Huan: Preliminaries and Overview, in: Dong, Guozhu; Liu, Huan (Hrsg.): Feature engineering for machine learning and data analytics, Chapman & Hall / CRC data mining & knowledge discovery series no. 44, CRC Press/Taylor & Francis Group, Boca Raton, FL, 2018

Eddy, N.: Machine Learning Drive (2016)
Eddy, Nathan: AI, Machine Learning Drive Autonomous Vehicle Development; https://www.informationweek.com/big-data/big-data-analytics/ai-machine-learning-drive-autonomous-vehicle-development/d/d-id/1325906?piddl_msgorder=thrd, 2016, Zugriff 19.09.2017

Edler, F. et al.: Fehlerbaumanalyse in Theorie und Praxis (2015)
Edler, Frank; Soden, Michael; Hankammer, René: Fehlerbaumanalyse in Theorie und Praxis, Imprint: Springer Vieweg, Berlin, Heidelberg, 2015

Elisseeff, A.; Pontil, M.: Leave-one-out error (2003)
Elisseeff, André; Pontil, Massimiliano: Leave-one-out error and stability of learning algorithms with applications, in: Suykens, J.A.K. et al. (Hrsg.): Advanced in Learning Theory: Methods, Models and Applications, NATO Science Series III: Computer and Systems Sciences Nr. 190, IOS Press, 2003

Everitt, B.; Skrondal, A.: The Cambridge dictionary of statistics (2010)
Everitt, Brian; Skrondal, Anders: The Cambridge dictionary of statistics, Finance professional collection, 4. Auflage, Cambridge University Press, Cambridge, New York, 2010

Evgeniou, T. et al.: Leave One Out Error (2004)
Evgeniou, Theodoros; Pontil, Massimiliano; Elisseeff, André: Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers, in: Machine Learning (1), Jahrgang 55, S. 71–97, 2004

Fakotakis, N.; Sgarbas, K. N.: Machine Learning in Human Language Technology (2001)
Fakotakis, Nikos; Sgarbas, Kyriakos N.: Machine Learning in Human Language Technology, in: Paliuras, Georgios (Hrsg.): Machine learning and its applications, Springer, Berlin [u.a.], 2001

Faria, J.: Machine Learning Safety (2018)
Faria, José: Machine Learning Safety, in: , 2018

Faria, J. M.: Non-determinism and Failure Modes in Machine Learning (2017)
Faria, Jose M.: Non-determinism and Failure Modes in Machine Learning, in: 2017 IEEE 28th International Symposium on Software Reliability Engineering workshops, IEEE, Piscataway, NJ, Piscataway, NJ, 2017

Figueroa, R. L. et al.: Predicting sample size required for classification performance (2012)
Figueroa, Rosa L.; Zeng-Treitler, Qing; Kandula, Sasikiran; Ngo, Long H.: Predicting sample size required for classification performance, in: BMC medical informatics and decision making, Jahrgang 12, S. 8, 2012

Filkovic, I.: Traffic Sign Localization and Classification Methods: An Overview (2014)
Filkovic, Ivan: Traffic Sign Localization and Classification Methods: An Overview University of Zagreb, Zagreb, 2014

Gebhardt, V.: Funktionale Sicherheit nach ISO 26262 (2013)
Gebhardt, Vera: Funktionale Sicherheit nach ISO 26262, dpunkt, Heidelberg, 2013

Goodfellow, I. J. et al.: Explaining and harnessing adversarial examples (2014)
Goodfellow, Ian J.; Shlens, Jonathon; Szegedy, Christian: Explaining and harnessing adversarial examples, in: arXiv preprint arXiv:1412.6572, 2014

Haltakov, V. et al.: Semantic Segmentation (2015)
Haltakov, Vladimir; Mayr, Jakob; Unger, Christian; Ilic, Slobodan: Semantic Segmentation Based Traffic Light Detection at Day and at Night, in: Gall, Juergen; Gehler, Peter; Leibe, Bastian (Hrsg.): Pattern Recognition, 1. Auflage, Springer International Publishing; Imprint: Springer, Cham, 2015

Hamalainen, W. et al.: Jerk-based feature extraction (2011)
Hamalainen, Wilhelmiina; Jarvinen, Mikko; Martiskainen, Paula; Mononen, Jaakko: Jerk-based feature extraction for robust activity recognition from acceleration data, in: 2011 11th International Conference on Intelligent Systems Design and Applications, IEEE, Piscataway, Nov. 2011

Hendricks, L. A. et al.: Generating Visual Explanations (2016)
Hendricks, Lisa A.; Akata, Zeynep; Rohrbach, Marcus; Donahue, Jeff; Schiele, Bernt; Darrell, Trevor: Generating Visual Explanations, in: Leibe, Bastian et al. (Hrsg.): Computer Vision ECCV 2016 [Elektronische Ressource], Imprint: Springer, Cham, 2016

Henze, N.: Die Binomialverteilung und die Multinomialverteilung (2018)
Henze, Norbert: Die Binomialverteilung und die Multinomialverteilung, in: Henze, Norbert (Hrsg.): Stochastik für Einsteiger, Springer Fachmedien Wiesbaden, Wiesbaden, 2018

Henzel, M. et al.: Herausforderungen in der Absicherung von FAS (2017)
Henzel, Maren; Winner, Hermann; Lattke, Benedikt: Herausforderungen in der Absicherung von Fahrerassistenzsystemen bei der Benutzung maschinell gelernter und lernender Algorithmen, in: Uni-DAS e.V. (Hrsg.): 11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, 2017

Hering, E.; Schönfelder, G.: Sensoren in Wissenschaft und Technik (2012)
Hering, Ekbert; Schönfelder, Gert: Sensoren in Wissenschaft und Technik, Vieweg+Teubner Verlag, Wiesbaden, 2012

Huang, Z.: Clustering Large Data Sets with Mixed Numeric and Categorical Values (1997)
Huang, Zhexue: Clustering Large Data Sets with Mixed Numeric and Categorical Values, in: The First Pacific-Asia Conference on Knowledge Discovery and Data Mining, 1997

ISO: ISO 26262:2011. Road vehicles: Functional safety (2011)
International Organization for Standardization: 1: ISO 26262:2011. Road vehicles: Functional safety, International Organization for Standardization, Geneva, 2011

ISO: ISO 26262:2018. Road vehicles: Functional safety (2018)
International Organization for Standardization: 2: ISO 26262:2018. Road vehicles: Functional safety, International Organization for Standardization, Geneva, 2018

ISO: My ISO job (2018)
International Organization for Standardization: My ISO job, 2018

ISO: Deliverables (2019)
International Organization for Standardization: Deliverables; https://www.iso.org/deliverables-all.html#IWA, 2019, Zugriff 07.05.2019

ISO: ISO/ PAS 21448 (2019)
International Organization for Standardization: 1: ISO/ PAS 21448. Road vehicles -- Safety of the intended functionality, 2019

ISO: ISO/ SAE CD 21434 (2019)
International Organization for Standardization: ISO/ SAE CD 21434. Road vehicles -- Cybersecurity engineering, 2019

Japkowicz, N.: Learning from imbalanced data sets (2000)
Japkowicz, Nathalie: Learning from imbalanced data sets, in: AAAI workshop on learning from imbalanced data sets, 2000

Jordan, J.: Evaluating a machine learning model (2017)
Jordan, Jeremy: Evaluating a machine learning model;

https://www.jeremyjordan.me/evaluating-a-machine-learning-model/, 2017, Zugriff 03.03.2019

Katz, G. et al.: Reluplex (2017)
Katz, Guy; Barrett, Clark; Dill, David; Julian, Kyle; Kochenderfer, Mykel: Reluplex; http://arxiv.org/pdf/1702.01135, 2017

Khurshudov, A.: Suddenly, a leopard print sofa appears (2015)
Khurshudov, Artem: Suddenly, a leopard print sofa appears; http://rocknrollnerd.github.io/ml/2015/05/27/leopard-sofa.html, 2015, Zugriff 30.01.2018

Kirkpatrick, C.; Dahlquist, J.: Technical Analysis (2010)
Kirkpatrick, Charles; Dahlquist, Julie: Technical Analysis, FT Press, 2010

Koopman, P.; Wagner, M.: Autonomous Vehicle Safety (2017)
Koopman, Philip; Wagner, Michael: Autonomous Vehicle Safety, in: IEEE Intelligent Transportation Systems Magazine (1), Jahrgang 9, S. 90–96, 2017

Kortenkamp, D.; Simmons, R.: Robotic Systems Architectures and Programming (2008)
Kortenkamp, David; Simmons, Reid: Robotic Systems Architectures and Programming, in: Siciliano, Bruno (Hrsg.): Springer handbook of robotics, Springer, Berlin [u.a.], 2008

Kramer, O.: Neuronale Netze (2009)
Kramer, Oliver: Neuronale Netze, in: Kramer, Oliver (Hrsg.): Computational Intelligence, Informatik im Fokus, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009

Kuhnl, T. et al.: Monocular road segmentation using slow feature analysis (2011)
Kuhnl, Tobias; Kummert, Franz; Fritsch, Jannik: Monocular road segmentation using slow feature analysis, in: IEEE Intelligent Vehicles Symposium (IV), 2011 ; 5 - 9 June 2011 ; Baden-Baden, Germany, IEEE, Piscataway, NJ, 2011

Kurd, Z.: Dissertation, Neural Networks in Safety-critical Applications (2002)
Kurd, Zeshan: Artificial Neural Networks in Safety-critical Applications, Dissertation University of York, York, 2002

Kurd, Z. et al.: Developing neural networks for safety critical systems (2006)
Kurd, Zeshan; Kelly, Tim; Austin, Jim: Developing artificial neural networks for safety critical systems, in: Neural Computing and Applications (1), Jahrgang 16, S. 11–19, 2006

Kurd, Z.; Kelly, T.: Establishing Safety Criteria for NN (2003)
Kurd, Zeshan; Kelly, Tim: Establishing Safety Criteria for Artificial Neural Networks, in: Palade, Vasile; Howlett, Robert J.; Jain, Lakhmi (Hrsg.): Knowledge-Based Intelligent Information and Engineering Systems [Elektronische Ressource], Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2003

Kwok, J. et al.: Machine Learning (2015)
Kwok, James; Zhou, Zhi-Hua; Xu, Lei: Machine Learning, in: Kacprzyk, Janusz; Pedrycz,

Witold (Hrsg.): Springer Handbook of Computational Intelligence edited by Janusz Kacprzyk, Witold Pedrycz, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015

Label (2017) Label, in: Sammut, Claude; Webb, Geoffrey I. (Hrsg.): Encyclopedia of Machine Learning and Data Mining, Springer US, Boston, MA, 2017

Li, L. et al.: Knows what it knows: a framework for self-aware learning (2008)
Li, Lihong; Littman, Michael L.; Walsh, Thomas J.: Knows what it knows: a framework for self-aware learning, in: Proceedings of the 25th international conference on Machine learning, 2008

Lisboa, P. J.: Industrial use of safety-related artificial neural networks (2001)
Lisboa, Paulo J. G.: Industrial use of safety-related artificial neural networks, HSE contract research reportno 327/2001, HSE Books, 2001

Loh, W.-Y.: Regression by Parts: Fitting Visually Interpretable Models with GUIDE (2008)
Loh, Wei-Yin: Regression by Parts: Fitting Visually Interpretable Models with GUIDE, in: Chen, Chun-houh; Härdle, Wolfgang; Unwin, Antony (Hrsg.): Handbook of data visualization, Springer handbooks of computational statistics, Springer, Berlin, London, 2008

Lombacher, J. et al.: Semantic radar grids (2017)
Lombacher, Jakob; Laudt, Kilian; Hahn, Markus; Dickmann, Jurgen; Wohler, Christian: Semantic radar grids, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017

Lubner, S.; Litzel, N.: Was ist ein Convolutional Neural Network? (2019)
Lubner, Stefan; Litzel, Nico: Was ist ein Convolutional Neural Network?; https://www.bigdata-insider.de/was-ist-ein-convolutional-neural-network-a-801246/, 2019, Zugriff 20.04.2019

Maiß, C.: Masterthesis, Literatur- und Patentrecherche maschinelles Lernen (2016)
Maiß, Christoph: Literatur- und Patentrecherche zu maschinellen Lernalgorithmen im Fahrzeug, Masterthesis
Technische Universität Darmstadt, Darmstadt, 2016

Mandalia, H. M.; Salvucci, M. D.: Using Support Vector Machines for Lane-Change (2016)
Mandalia, Hiren M.; Salvucci, Mandalia D. D.: Using Support Vector Machines for Lane-Change Detection, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting (22), Jahrgang 49, S. 1965–1969, 2016

Mandava, s.: Cross Validation and HyperParameter Tuning in Python (2018)
Mandava, saranya: Cross Validation and HyperParameter Tuning in Python; https://medium.com/@mandava807/cross-validation-and-hyperparameter-tuning-in-python-65cfb80ee485, 2018, Zugriff 06.03.2019

Marina Martinez, C. et al.: Driving Style Recognition for Intelligent Vehicle Control (2018)

Marina Martinez, Clara; Heucke, Mira; Wang, Fei-Yue; Gao, Bo; Cao, Dongpu: Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance, in: IEEE Transactions on Intelligent Transportation Systems (3), Jahrgang 19, S. 666–676, 2018

Marr, B.: The Amazing Ways Tesla Is Using Artificial Intelligence And Big Data (2018)
Marr, Bernard: The Amazing Ways Tesla Is Using Artificial Intelligence And Big Data; https://www.forbes.com/sites/bernardmarr/2018/01/08/the-amazing-ways-tesla-is-using-artificial-intelligence-and-big-data/, 2018, Zugriff 06.12.2018

MATLAB: Financial Toolbox Documentation
MATLAB: Financial Toolbox Documentation; https://de.mathworks.com/help/finance/tsmovavg.html#bt1732z-1-numperiod, Zugriff 29.03.2019

Mattmann, I.: Dissertation, Modellintegrierte Produkt- und Prozessentwicklung (2017)
Mattmann, Ilyas: Modellintegrierte Produkt- und Prozessentwicklung, Dissertation Technische Universität Darmstadt, Darmstadt, 2017

Maurer, M. et al.: Autonomes Fahren (2015)
Maurer, Markus; Gerdes, J. C.; Lenz, Barbara; Winner, Hermann (Hrsg.) Autonomes Fahren, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015

McDonald, C.: Machine learning fundamentals (I): Cost functions and gradient descent (2017)
McDonald, Conor: Machine learning fundamentals (I): Cost functions and gradient descent; https://towardsdatascience.com/machine-learning-fundamentals-via-linear-regression-41a5d11f5220, 2017, Zugriff 01.03.2019

Mirchevska, B. et al.: Reinforcement Learning for Autonomous Maneuvering (2017)
Mirchevska, Branka; Blum, Manuel; Louis, Lawrence; Boedecker, Joschka; Werling, Moritz: Reinforcement Learning for Autonomous Maneuvering in Highway Scenarios, in: Uni-DAS e.V. (Hrsg.): 11. Workshop Fahrerassistenzysteme und automatisiertes Fahren, Darmstadt, 2017

Mitchell, T. M.: Machine learning (1997)
Mitchell, Tom M.: Machine learning, McGraw-Hill series in computer science: Artificial Intelligence, McGraw-Hill, New York, NY, 1997

Morgun, I.: Types of machine learning algorithms (2015)
Morgun, Ivan: Types of machine learning algorithms; https://en.proft.me/2015/12/24/types-machine-learning-algorithms/, 2015, Zugriff 14.01.2019

Morik, K.: LS 8 Report 1, Maschinelles Lernen (1993)
Morik, Katharina: Maschinelles Lernen, LS 8 Report 1
Universität Dortmund, Dortmund, 1993

Moving Average (2008) Moving Average, in: Dodge, Yadolah (Hrsg.): The Concise Encyclopedia of Statistics, Springer New York, New York, NY, 2008

Mubaris: K-Means Clustering in Python (2017)
Mubaris: K-Means Clustering in Python; https://mubaris.com/posts/kmeans-clustering/, 2017, Zugriff 03.03.2019

Mukherjee, S. et al.: Estimating dataset size requirements (2003)
Mukherjee, Sayan; Tamayo, Pablo; Rogers, Simon; Rifkin, Ryan; Engle, Anna; Campbell, Colin; Golub, Todd R.; Mesirov, Jill P.: Estimating dataset size requirements for classifying DNA microarray data, in: Journal of computational biology (2), Jahrgang 10, S. 119–142, 2003

Mukherjee, U.: How to handle Imbalanced Classification Problems (2017)
Mukherjee, Upasana: How to handle Imbalanced Classification Problems in machine learning?; https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/, 2017, Zugriff 13.02.2018

Murphey, Y. L. et al.: Driver's style classification using jerk analysis (2009)
Murphey, Yi L.; Milton, Robert; Kiliaris, Leonidas: Driver's style classification using jerk analysis, in: 2009 IEEE Symposium on Computational Intelligence in Vehicles and Vehicular Systems (CIVVS 2009) proceedings, IEEE, [Piscataway, N.J.], 2009

Navarro, P. J. et al.: Pedestrian Detection for Autonomous Vehicles (2016)
Navarro, Pedro J.; Fernández, Carlos; Borraz, Raúl; Alonso, Diego: A Machine Learning Approach to Pedestrian Detection for Autonomous Vehicles Using High-Definition 3D Range Data, in: Sensors (Basel, Switzerland) (1), Jahrgang 17, 2016

Nusser, S.: Dissertation, Robust Learning in Safety-Related Domains (2009)
Nusser, Sebastian: Robust Learning in Safety-Related Domains, Dissertation
Otto-von-Guericke-Universität, Magdeburg, 2009

nvidia: NVIDIA Announces World’s First Functionally Safe AI Self-Driving Platform (2018)
nvidia: NVIDIA Announces World’s First Functionally Safe AI Self-Driving Platform; https://nvidianews.nvidia.com/news/nvidia-announces-worlds-first-functionally-safe-ai-self-driving-platform, 2018, Zugriff 17.02.2019

Olah, C.: Visualizing MNIST: An Exploration of Dimensionality Reduction (2014)
Olah, Chris: Visualizing MNIST: An Exploration of Dimensionality Reduction; http://colah.github.io/posts/2014-10-Visualizing-MNIST/, 2014

Olah, C. et al.: Feature Visualization (2017)
Olah, Chris; Mordvintsev, Alexander; Schubert, Ludwig: Feature Visualization, in: Distill (11), Jahrgang 2, 2017

Otte, C.: Safe and Interpretable Machine Learning (2013)
Otte, Clemens: Safe and Interpretable Machine Learning: A Methodological Review, in: Moewes, Christian; Nürnberger, Andreas (Hrsg.): Computational Intelligence in Intelligent Data Analysis, Studies in Computational Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013

Papernot, N. et al.: Technical Report on the CleverHans v2.1.0 Adversarial Examples Library (2016)
Papernot, Nicolas; Faghri, Fartash; Carlini, Nicholas; Goodfellow, Ian; Feinman, Reuben; Kurakin, Alexey; Xie, Cihang; Sharma, Yash; Brown, Tom; Roy, Aurko; Matyasko, Alexander; Behzadan, Vahid; Hambardzumyan, Karen; Zhang, Zhishuai; Juang, Yi-Lin; Li, Zhi; Sheatsley, Ryan; Garg, Abhibhav; Uesato, Jonathan; Gierke, Willi; Dong, Yinpeng; Berthelot, David; Hendricks, Paul; Rauber, Jonas; Long, Rujun; McDaniel, Patrick: Technical Report on the CleverHans v2.1.0 Adversarial Examples Library, 2016

Park, J. et al.: Intelligent Energy Management and Optimization (2016)
Park, Jungme; Murphey, Yi L.; Abul Masrur, M.: Intelligent Energy Management and Optimization in a Hybridized All-Terrain Vehicle With Simple On–Off Control of the Internal Combustion Engine, in: IEEE Transactions on Vehicular Technology (6), Jahrgang 65, S. 4584–4596, 2016

Plaza-Leiva, V. et al.: Classification of Lidar Point Clouds (2017)
Plaza-Leiva, Victoria; Gomez-Ruiz, Jose A.; Mandow, Anthony; García-Cerezo, Alfonso: Voxel-Based Neighborhood for Spatial Shape Pattern Classification of Lidar Point Clouds with Supervised Learning, in: Sensors (Basel, Switzerland) (3), Jahrgang 17, 2017

Pomerleau, D. A.: ALVINN (1989)
Pomerleau, Dean A.: ALVINN, in: Touretzky, D. S. (Hrsg.): Advances in Neural Information Processing Systems 1, Morgan-Kaufmann, 1989

Priese, L.: Computer Vision (2015)
Priese, Lutz (Hrsg.) Computer Vision, eXamen.press, 1. Auflage, Springer Berlin Heidelberg; Springer Berlin; Springer Vieweg, [s.l.], Berlin, [s.l.], 2015

Ragland, D. R. et al.: Gap acceptance for vehicles turning left across on-coming traffic
Ragland, David R.; Arroyo, Sofia; Shladover, Steven E.; Misener, James A.; Chan, Ching-Yao: Gap acceptance for vehicles turning left across on-coming traffic
UC Berkeley

Ramachandran, U. B.: Masterthesis, Issues in Verification and Validation of NN (2005)
Ramachandran, Uma B.: Issues in Verification and Validation of Neural Network Based

Approaches for Fault-Diagnosis in Autonomous Systems, Masterthesis
Concordia University, Montreal, 2005

Ramasubramanian, K.; Singh, A.: Machine Learning Using R (2017)
Ramasubramanian, Karthik; Singh, Abhishek: Machine Learning Using R, Apress, Berkeley, CA, 2017

Rausch, S.: Master-Thesis, Ableitung einer Simulationsumgebung aus der realen Welt (2016)
Rausch, Sebastian: Entwicklung einer Methode zur Ableitung einer Simulationsumgebung aus der realen Welt für maschinelle Lernverfahren, Master-Thesis
Technische Universität Darmstadt, Darmstadt, 2016

Rebhan, S.; Kleinehagenbrock, M.: Intelligent gap setting (2016)
Rebhan, Sven; Kleinehagenbrock, Marcus: Intelligent gap setting for adaptive cruise control, Honda Research Institute Europe GmH, Patent EP3081447A1, Patent Anmeldenummer: EP20150163486, 2016

Rezaei, M.; Klette, R.: Computer Vision for Driver Assistance (2017)
Rezaei, Mahdi; Klette, Reinhard: Computer Vision for Driver Assistance, Jahrgang 45, Springer International Publishing, Cham, 2017

Rosenfeld, A. et al.: Improve the Acceptance of ACC (2012)
Rosenfeld, Avi; Bareket, Zevi; Goldman, Claudia V.; Kraus, Sarit; LeBlanc, David J.; Tsimhoni, Omer: Learning Driver's Behavior to Improve the Acceptance of Adaptive Cruise Control, in: IAAI, 2012

Rudolph, A. et al.: A consistent safety case argumentation for artificial intelligence (2018)
Rudolph, Alexander; Voget, Stefan; Mottok, Jürgen: A consistent safety case argumentation for artificial intelligence in safety related automotive systems, in: Embedded Real Time Software and Systems, 2018

Rüger, F. et al.: Kontrollierbarkeitsbewertung von FAS (2015)
Rüger, Fabian; Sieber, Markus; Siegel, Andreas; Siederberger Karl-Heinz; Färber. Berthold: Kontrollierbarkeitsbewertung von FAS der aktiven Sicherheit in frühen Phasen des Entwicklungsprozesses mit dem Vehicle in the Loop (VIL), in: Uni-DAS e.V. (Hrsg.): 9. Workshop Fahrerassistenzsysteme, Darmstadt, 2015

Said, C.: Driving the future (2017)
Said, Carolyn: Driving the future:, in: San Francisco Chronicle, Jahrgang 2017, 2017

Salay, R. et al.: An Analysis of ISO 26262 (2017)
Salay, Rick; Queiroz, Rodrigo; Czarnecki, Krzysztof: An Analysis of ISO 26262; http://arxiv.org/pdf/1709.02435, 2017

Salay, R.; Czarnecki, K.: Using Machine Learning Safely in Automotive Software (2018)

Salay, Rick; Czarnecki, Krzysztof: Using Machine Learning Safely in Automotive Software, 2018

Sammut, C.; Webb, G. I.: Encyclopedia of Machine Learning and Data Mining (2017)
Sammut, Claude; Webb, Geoffrey I. (Hrsg.) Encyclopedia of Machine Learning and Data Mining, Springer US, Boston, MA, 2017

Schmitt, W.: Verfahren zum Lernen von Synchronisationsschwellen (2016)
Schmitt, Werner: Verfahren zum Lernen von Synchronisationsschwellen eines Doppelkupplungsgetriebes in einem Kraftfahrzeug, Schaeffler Technologies AG & Co. KG, Patent DE102015220918 A1, 2016

Schnieder, L.; Hosse, R. S.: Leitfaden Safety of the Intended Functionality (2019)
Schnieder, Lars; Hosse, René S.: Leitfaden Safety of the Intended Functionality, Springer Fachmedien Wiesbaden, Wiesbaden, 2019

scikit-learn: 2.3. Clustering — scikit-learn 0.20.3 documentation (2019)
scikit-learn: 2.3. Clustering — scikit-learn 0.20.3 documentation; https://scikit-learn.org/stable/modules/clustering.html#k-means, 2019, Zugriff 18.03.2019

scikit-learn: sklearn.cluster.KMeans (2019)
scikit-learn: sklearn.cluster.KMeans; https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans, 2019, Zugriff 06.03.2019

Sebe, N. et al.: Machine Learning in Computer Vision (2005)
Sebe, Nicu; Cohen, Ira; Garg, Ashutosh; Huang, Thomas: Machine Learning in Computer Vision, Jahrgang 29, Springer-Verlag, Berlin/Heidelberg, 2005

Seeger, C. et al.: Towards road type classification with occupancy grids (2016)
Seeger, Christoph; Muller, A.; Schwarz, Loren; Manz, Michael: Towards road type classification with occupancy grids, in: IEEE Intelligent Vehicles Symposium 2016 Workshop: DeepDriving-Learning Representations for Intelligent Vehicles, 2016

Shah, T.: About Train, Validation and Test Sets in Machine Learning (2017)
Shah, Tarang: About Train, Validation and Test Sets in Machine Learning; https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7, 2017, Zugriff 16.01.2019

Singer, C.: Dissertation, Entwicklung von Testauswahlmethoden (2015)
Singer, Christina: Entwicklung von Testauswahlmethoden für die Absicherung von Änderungen auf Gesamtfahrzeugebene, Dissertation
Technische Universität Darmstadt, Darmstadt, 2015

Smart Eye AB: Technology | Smart Eye (2016)
Smart Eye AB: Technology | Smart Eye; http://smarteye.se/technology/, 2016, Zugriff 26.07.2017

Spanfelner, B. et al.: Challenges in applying the ISO 26262 (2012)
Spanfelner, Bernd; Richter, Detlev; Ebel, Susanne; Wilhelm, Ulf; Branz, Wolfgang; Patz, Carsten: Challenges in applying the ISO 26262 for driver assistance systems, in: Technische Universität München (Hrsg.): 5. Tagung Fahrerassistenz, München, 2012

Stallkamp, J. et al.: Man vs. computer (2012)
Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C.: Man vs. computer, in: Neural networks : the official journal of the International Neural Network Society, Jahrgang 32, S. 323–332, 2012

Stein, G.: Bundling of driver assistance systems (2010)
Stein, Gideon: Bundling of driver assistance systems, Mobileye Vision Technologies, Patent EP2172873 A2, Patent Anmeldenummer: P20090252361, 2010

Szegedy, C. et al.: Intriguing properties of neural networks (2013)
Szegedy, Christian; Zaremba, Wojciech; Sutskever, Ilya; Bruna, Joan; Erhan, Dumitru; Goodfellow, Ian; Fergus, Rob: Intriguing properties of neural networks, in: arXiv preprint arXiv:1312.6199, 2013

Taylor, B. J. et al.: Verification and validation of neural networks (2003)
Taylor, Brian J.; Darrah, Marjorie A.; Moats, Christina D.: Verification and validation of neural networks, in: Priddy, Kevin L.; Angeline, Peter J. (Hrsg.): , SPIE Proceedings, SPIE, 2003

Tch, A.: The mostly complete chart of Neural Networks, explained (2017)
Tch, Andrew: The mostly complete chart of Neural Networks, explained; https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464, 2017, Zugriff 20.04.2019

Tesla: Software-Updates (2019)
Tesla: Software-Updates; https://www.tesla.com/de_DE/support/software-updates, 2019, Zugriff 25.02.2019

test IO: Black Box Testing (2019)
test IO: Black Box Testing; https://test.io/black-box-testing/, 2019, Zugriff 20.04.2019

The Assurance Case Working Group: Goal Structuring Notation (2018)
The Assurance Case Working Group: Goal Structuring Notation Community Standard, 2. Auflage, 2018

Thoma, M.: A survey of semantic segmentation (2016)
Thoma, Martin: A survey of semantic segmentation, in: arXiv preprint arXiv:1602.06541, 2016

Thrun, S.: Bayesian Landmark Learning (1998)
Thrun, Sebastian: Bayesian Landmark Learning for Mobile Robot Localization, in: Machine Learning (1), Jahrgang 33, S. 41–76, 1998

Trevino, A.: Introduction to K-means Clustering (2016)
Trevino, Andrea: Introduction to K-means Clustering;
https://www.datascience.com/blog/k-means-clustering, 2016, Zugriff 03.03.2019

V, A. S.: Understanding Activation Functions in Neural Networks (2017)
V, Avinash S.: Understanding Activation Functions in Neural Networks;
https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0, 2017, Zugriff 20.04.2019

Varshney, K. R.: Engineering safety in machine learning (2016)
Varshney, Kush R.: Engineering safety in machine learning, in: 2016 Information Theory and Applications Workshop (ITA), IEEE, 2016

Viégas, F.; Wattenberg, M.: Visualization for Machine Learning (2018)
Viégas, Fernanda; Wattenberg, Martin: Visualization for Machine Learning, in: 32st Annual Conference on Neural Information Processing Systems, 2018

Viehof, M.: Dissertation, Objektive Qualitätsbewertung von Fahrdynamiksimulationen (2018)
Viehof, Michael: Objektive Qualitätsbewertung von Fahrdynamiksimulationen durch statistische Validierung, Dissertation
Technische Universität Darmstadt, Darmstadt, 2018

Viola, P.; Jones, M. J.: Robust Real-Time Face Detection (2004)
Viola, Paul; Jones, Michael J.: Robust Real-Time Face Detection, in: International Journal of Computer Vision (2), Jahrgang 57, S. 137–154, 2004

Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015)
Wachenfeld, Walther; Winner, Hermann: Die Freigabe des autonomen Fahrens, in: Maurer, Markus et al. (Hrsg.): Autonomes Fahren, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015

Wachenfeld, W.; Winner, H.: Lernen autonome Fahrzeuge? (2015)
Wachenfeld, Walther; Winner, Hermann: Lernen autonome Fahrzeuge?, in: Maurer, Markus et al. (Hrsg.): Autonomes Fahren, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015

Walch, F.: Masterthesis, Deep Learning for Image-Based Localization (2016)
Walch, Florian: Deep Learning for Image-Based Localization, Masterthesis
Technische Universität München, München, 2016

Wang, F.; Rudin, C.: Causal Falling Rule Lists (2017)
Wang, Fulton; Rudin, Cynthia: Causal Falling Rule Lists; http://arxiv.org/pdf/1510.05189, 2017

Wang, R.; Lukic, S. M.: Review of driving conditions prediction and driving style recognition (2011)
Wang, Rui; Lukic, Srdjan M.: Review of driving conditions prediction and driving style

recognition based control algorithms for hybrid electric vehicles, in: Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE, [IEEE], [Piscataway, N.J.], 2011

Wang, W. et al.: Modeling and Recognizing Driver Behavior Based on Driving Data (2014)
Wang, Wenshuo; Xi, Junqiang; Chen, Huiyan: Modeling and Recognizing Driver Behavior Based on Driving Data, in: Mathematical Problems in Engineering (1), Jahrgang 2014, S. 1–20, 2014

Weitzel, A. et al.: Absicherungsstrategien für Fahrerassistenzsysteme (2014)
Weitzel, Alexander; Winner, Hermann; Peng, Cao; Geyer, Sebastian; Lotz, Felix; Sefati, Mohsen: Absicherungsstrategien für Fahrerassistenzsysteme mit Umfeldwahrnehmung, Berichte der Bundesanstalt für Strassenwesen - Fahrzeugtechnik (F), Jahrgang 98, Wirtschaftsverl. NW Verl. für neue Wissenschaft, Bremerhaven, 2014

Wilhelm, U. et al.: Functional Safety of Driver Assistance Systems and ISO 26262 (2016)
Wilhelm, Ulf; Ebel, Susanne; Weitzel, Alexander: Functional Safety of Driver Assistance Systems and ISO 26262, in: Winner, Hermann et al. (Hrsg.): Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort, Springer International Publishing, Cham, 2016

Winner, H.: Handbuch Fahrerassistenzsysteme (2015)
Winner, Hermann (Hrsg.) Handbuch Fahrerassistenzsysteme, Vieweg+Teubner Verlag; Wiesbaden : Imprint: Springer Vieweg, Wiesbaden, 2015

Winner, H. et al.: Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort (2016)
Winner, Hermann; Hakuli, Stephan; Lotz, Felix; Singer, Christina (Hrsg.) Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort, Springer International Publishing, Cham, 2016

Winter, J. C.F. de **et al.: Left turn gap acceptance in a simulator (2010)**
Winter, J. C.F. de; Spek, A. C.E.; Groot, S. de; Wieringa, P. A.: Left turn gap acceptance in a simulator, in: Proceedings Driving Simulation Conference 2009, 2010

Yosinski, J. et al.: Understanding Neural Networks (2015)
Yosinski, Jason; Clune, Jeff; Nguyen, Ahn; Fuchs, Thomas; Hod, Lipson: Understanding Neural Networks Through Deep Visualization, in: Deep Learning Workshop Nr. 32, Lille, 2015

Zhang, Q. et al.: Examining CNN Representations with respect to Dataset Bias (2017)
Zhang, Quanshi; Wang, Wenguan; Zhu, Song-chun: Examining CNN Representations with respect to Dataset Bias, 2017

Zhang, Q.-s.; Zhu, S.-c.: Visual interpretability for Deep Learning (2018)
Zhang, Quan-shi; Zhu, Song-chun: Visual interpretability for Deep Learning, in: Frontiers of Information Technology & Electronic Engineering (1), Jahrgang 19, S. 27–39, 2018

Zhang, Y. et al.: Learning-Based Driver Workload Estimation (2008)
Zhang, Yilu; Owechko, Yuri; Zhang, Jing: Learning-Based Driver Workload Estimation, in: Prokhorov, Danil (Hrsg.): Computational Intelligence in Automotive Applications, Studies in Computational Intelligence Nr. 132, 1. Auflage, Springer Berlin, Berlin, 2008

Eigene Veröffentlichungen

Betz, A.; Winner, H.; Ancochea, M.; Graupner, M.: Motion Analysis of a Wheeled Mobile Driving Simulator for Urban Traffic Situations, Driving Simulator Conference 2012, September 6-7, Paris, France, 2012.

Henzel, M; Winner, H.; Lattke, B.: Herausforderungen in der Absicherung von Fahrerassistenzsystemen bei der Benutzung maschinell gelernter und lernender Algorithmen, 11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, 29. – 31. März 2017, Walting, 2017.

Fecher, N.; Graupner, M.; Winner, H.: Wirksamkeitsbewertung im realen Fahrversuch, Tagung „Fahrerassistenz und Aktive Sicherheit“, Haus der Technik, Essen, 16.-17.04.2015.

Winner, H.; Graupner, M.: PROMETHEUS – Welche Visionen wurden zur Realität? In: Tagungsband 17. VDA Technischer Kongress, Filderstadt, 19. und 20. März 2015.

Winner, H.; Graupner, M.; Wachenfeld, W.: How to Address the Approval Trap for Autonomous Vehicles. In: Vortrag ITSC (Keynote). September 17, Gran Canaria, 2015.

Betreute studentische Arbeiten

Frieder Gottmann, Maschinenbau: Entwicklung einer Softwarearchitektur für ein Conduct-by-Wire-Versuchsfahrzeug. Master-Thesis Nr. 559/14, 2015

Marc Hetschger, Wirtschaftsingenieurwesen Maschinenbau: Hedonische Qualität in Fahrerassistenzsystemen. Master-Thesis Nr. 625/16, 2017

Nicolas Jourdan, Maschinenbau: Erkennung von Unsicherheit und Anomalien in Künstlichen Neuronalen Netzen. Master-Thesis Nr. 715/18, 2019

Nicolas Nostadt, Maschinenbau: Entwicklung einer Methode zur Bestimmung von Fahrstilen anhand von Fahrzeug- und Umweltgrößen. Master-Thesis Nr. 621/16, 2017

David Maiß, Wirtschaftsingenieurwesen Maschinenbau: Literatur- und Patentrecherche zu maschinellen Lernalgorithmen im Fahrzeug. Master-Thesis Nr. 624/16, 2017

Jalal Mirzayev, Maschinenbau: Maschinelles Lernen einer automatisierten Fahraufgabe durch Nachahmung. Master-Thesis Nr. 721/18, 2019

Sebastian Rausch, Maschinenbau: Entwicklung einer Methode zur Ableitung einer Simulationsumgebung aus der realen Welt für maschinelle Lernverfahren. Master-Thesis Nr. 601/15, 2016

Yannick Ryma, Maschinenbau: Konzeptionierung und Implementierung eines aktiven Bußgeldkatalogs. Master-Thesis Nr. 626/16, 2017